

Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки

Рівень вищої освіти другий (магістерський)

Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка

(код і повна назва)

Тип програми Освітньо-професійна

Освітня програма Комп'ютеризовані та робототехнічні системи

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« ____ » _____ 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Стеценко Катерині Костянтинівні

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення робота-компаньйона для людей з обмеженими можливостями

затверджена наказом університету від 10.11.2025 р. № 1018 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 11.12. 2025 р.

3. Вихідні дані до роботи Пусковий струм = 2А, напруга для мобільної платформи = 7,4 В, ультразвукові датчики HC-SR04, плата керування = ESP32 DevKit V4, роздільна здатність 5 = Мп.

4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ; 4.2 Аналіз сучасних інтерактивних роботів-помічників; 4.3 Розробка макету робота-помічника; 4.4 Розробка програми робота-помічника; 4.5 Експериментальні дослідження та аналіз отриманих результатів; 4.6 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Демонстраційний матеріал, представлений у форматі презентації PowerPoint (*.ppt). – 14 с.ф. А 4.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз сучасних інтерактивних роботів-помічників	01.09.2025	виконано
2	Розробка макету робота-помічника	10.09.2025	виконано
3	Розробка програми робота-помічника	12.10.2025	виконано
4	Експериментальні дослідження та аналіз отриманих результатів	14.10.2025	виконано
5	Оформлення пояснювальної записки	18.11.2025	виконано
6	Подання роботи на рецензію	31.11.2025	виконано
7	Подання роботи на підпис зав.кафедри	11.12.2025	

Дата видачі завдання 01.09.2025 р.

Здобувач _____
(підпис)

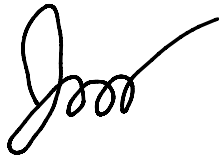
Керівник роботи _____
(підпис)

Катерина СТЕЦЕНКО
(власне ім'я, прізвище)

доц. Артем БРОННІКОВ
(посада, власне ім'я, прізвище)

Я, здобувачка вищої освіти ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавала і не одержувала недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

25 листопада 2025 р.



Катерина СТЕЦЕНКО

РЕФЕРАТ

Пояснювальна записка: 87 с., 3 табл., 42 рис., 3 дод., 34 джерела.

РОБОТ ПОМІЧНИК ДЛЯ ЛЮДЕЙ З ОБМЕЖЕНИМИ
МОЖЛИВОСТЯМИ, ШТУЧНИЙ ІНТЕЛЕКТ, МОБІЛЬНА ПЛАТФОРМА З
МАНІПУЛЯТОРОМ.

Мета роботи – підвищення ефективності функціонування інтелектуальної системи для людей з обмеженими можливостями шляхом створення робота-помічника.

Об'єкт дослідження – процес керування мобільним роботом-компаньйоном з використанням технологій комп'ютерного зору та розпізнавання мовлення.

Предмет дослідження – система прийняття рішень мобільного робота-компаньйона на основі сенсорної інформації.

В кваліфікаційній роботі було проведено аналіз сучасних технологічних рішень та реалізацій роботів-помічників для людей з обмеженими можливостями, розглянуто конструкторські особливості та проаналізовано алгоритм взаємодії з користувачем. В результаті аналізу було прийнято рішення інтегрувати до помічника штучний інтелект для розпізнавання мови, генерування відповідей та можливості надання психологічної підтримки.

Розроблено структурну схему, схему підключення та було обрано апаратні модулі, що задовольняють умови. На основі проведеної роботи було зібрано макет робота. Розроблено алгоритм роботи голосового помічника та мобільного робота з механічною рукою.

Проведено експериментальне дослідження процесу розпізнавання роботом контексту та коректності відповідей. Для платформи було проведено дослідження здатності орієнтуватися в просторі та запобігати перешкод.

Підтримуючи проголошені резолюцією Генеральної Асамблеї Організації Об'єднаних Націй від 25 вересня 2015 року № 70/1 глобальні цілі сталого розвитку до 2030 року та результати їх адаптації з урахуванням специфіки розвитку України, що викладені у Національній доповіді «Цілі сталого розвитку: Україна», надані результати дослідження забезпечують дотримання Цілей сталого розвитку України на період до 2030 року. Вони узгоджуються з Ціллю 3 «Забезпечення здорового способу життя та сприяння благополуччю для всіх у будь-якому віці», зокрема з пунктом 3.9 – «Здійснити реформу фінансування системи охорони здоров'я», бо розроблений робот сприяє покращенню якості життя багатьом людям з обмеженими фізичними можливостями

ABSTRACT

The explanatory note contains 87 p., 3 tables, 42 drawings, 3 pp., 31 sources.

ROBOT ASSISTANT FOR PEOPLE WITH DISABILITIES, ARTIFICIAL INTELLIGENCE, MOBILE PLATFORM WITH MANIPULATOR.

The goal of this work is to improve the effectiveness of the intellectual system for people with disabilities by creating a robot assistant.

The object of the study is the process of controlling a mobile companion robot using computer vision and speech recognition technologies.

The subject of the study is the decision-making system of a mobile companion robot based on sensory information.

The thesis analyzes modern technological solutions and implementations of assistant robots for people with disabilities, considers design features, and analyzes the algorithm of interaction with the user. As a result of the analysis, it was decided to integrate artificial intelligence into the assistant for speech recognition, response generation, and the ability to provide psychological support.

A structural diagram and connection diagram were developed, and hardware modules that meet the requirements were selected. Based on the work done, a model of the robot was assembled. An algorithm was developed for the voice assistant and mobile robot with a mechanical arm.

An experimental study was conducted on the robot's understanding of context and the appropriateness of responses. Research was conducted on the platform's ability to navigate space and avoid obstacles.

Supporting the global sustainable development goals for 2030 proclaimed by the United Nations General Assembly Resolution No. 70/1 of September 25, 2015, and the results of their adaptation to the specifics of Ukraine's development, as set out in the National Report "Sustainable Development Goals: Ukraine," the results of the

study ensure compliance with Ukraine's Sustainable Development Goals for the period up to 2030. They are consistent with Goal 3, “Ensure healthy lives and promote well-being for all at all ages,” in particular with paragraph 3.9, “Reform health financing systems,” because the developed robot contributes to improving the quality of life for many people with physical disabilities.

ЗМІСТ

Перелік умовних скорочень.....	9
Вступ.....	10
1 Аналіз сучасних інтерактивних роботів-помічників.....	12
1.1 Аналіз конструкцій та апаратної архітектури інтерактивних роботів-компаньйонів.....	13
1.2 Аналіз систем управління та взаємодії інтерактивних роботів-компаньйонів з користувачами	17
1.2.1 Порівняльний аналіз сучасних роботів-компаньйонів	18
1.2.2 Системи прийняття рішень у роботах-компаньйонах	21
1.3 Аналіз технологій адаптивної підтримки та персоналізації для людей з обмеженими можливостями	23
1.3.1 Прийняття рішень у системах автономних мобільних роботів.....	25
2 Розробка макету робота-помічника.....	28
2.1 Розробка загальної структурної схеми робота-помічника	28
2.2 Вибір апаратних модулів	30
2.3 Розробка схеми підключення.....	35
2.4 Розрахунок системи управління	38
2.5 Моделювання робота в середовищі Tinkercad	42
2.6 Збірка макету робота	44
2.7 Вибір середовища для програмування робота помічника.....	49
3. Розробка програми робота-помічника	53
3.1 Розробка загального алгоритму роботи	53
3.2 Реалізація розпізнавання об'єктів алгоритмом YOLO	57
3.3 Реалізація розумного помічника	61
3.4 Розрахунок параметрів електробезпеки та заземлювальних пристроїв	66
4 Експериментальні дослідження та аналіз отриманих результатів	69
4.1 Постановка задач експерименту з запросами до ChatGPT	69
4.2 Дослідження стабільності відповіді	73

	8
4.3 Дослідження об'їзду перешкод мобільним роботом	74
4.4 Висновки з експериментів в 4-му розділі.....	76
Висновки.....	78
Перелік джерел посилання	81
Додаток А Лістинг програми для керування роботом помічником	86
Додаток Б Результати експерименту	98
Додаток В Публікації.....	5
Додаток Г Демонстраційний матеріал	21

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БПЛА – безпілотні літальні апарати;

API – Application Programming Interface;

DQN – Deep Q-Networks;

HSR – Human Support Robot;

LLM – Large Language Models;

MDP – Markov Decision Processes;

MP – Mobile Platform;

RL – Reinforcement Learning;

WoZ – Wizard of Oz;

XOLO – You Only Look Once.

ВСТУП

В час коли технології швидко розвиваються та допомагають поліпшити якість життя людини, одним з ключових напрямків є розробка роботів-компаньйонів або ж роботів-помічників, які виконують не лише базові функції, а й забезпечують психологічну та соціальну підтримку.

Особливої актуальності проблема набуває в умовах повномасштабного вторгнення, все більше військових втрачають частину кінцівок або ж значну частину тіла. Для таких людей є критичною проблемою не мати змоги повернутися до нормального життя, адаптуватися, але роботи можуть надати підтримку в побутових завданнях, психологічно-соціальних аспектах.

Роботи-помічники для людей з обмеженими можливостями вирішують дані проблеми завдяки інтегрування штучного інтелекту. Робот може аналізувати, мати пам'ять, обробляти користувацькі запити та виконувати роль партнера у виконанні утилітарних функцій.

Мета роботи – підвищення ефективності функціонування інтелектуальної системи для людей з обмеженими можливостями шляхом створення робота-помічника, здатного до автономного функціонування та контекстно-залежної взаємодії з користувачем.

Об'єкт дослідження – процес керування мобільним роботом-компаньйоном з використанням технологій комп'ютерного зору та розпізнавання мовлення.

Предмет дослідження – система прийняття рішень мобільного робота-компаньйона на основі сенсорної інформації.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– аналіз конструкцій та апаратної архітектури інтерактивних роботів-компаньйонів;

- аналіз систем управління та взаємодії інтерактивних роботів-компаньйонів з користувачами;
- аналіз технологій адаптивної підтримки та персоналізації для людей з обмеженими можливостями;
- розробка загальної структурної схеми робота-помічника;
- вибір апаратних модулів та розробка схеми підключення;
- збірка макету робота;
- розробка загального алгоритму роботи;
- реалізація розумного помічника;
- дослідження стабільності відповіді та об'їзду перешкод мобільним роботом.

Кваліфікаційна робота виконана згідно ДСТУ 3008–15 [1] та методичних вказівок [2], із використанням статей та матеріалів із наукових публікацій. Та проведено апробацію у [3] та [4].

1 АНАЛІЗ СУЧАСНИХ ІНТЕРАКТИВНИХ РОБОТІВ-ПОМІЧНИКІВ

Сучасний етап інтенсивного розвитку суспільства характеризується значущими змінами в демографії суспільства, що створюють принципово нові задачі для систем соціального забезпечення та медичного догляду. Процеси старіння населення, зниження показників народжуваності та збільшення кількості людей з обмеженими функціональними можливостями формують структурний дисбаланс між зростаючими потребами у послугах по догляду та наявними людськими ресурсами для їх забезпечення.

Особливу увагу для вирішення проблем приділяють у високорозвинутих країнах. Зокрема у Японії, де демографічна криза досягла рівня, що загрожує стабільності соціального забезпечення. Дефіцит кваліфікованих спеціалістів у сфері догляду за людьми з пільгами актуалізує потребу в розробці альтернативних технологічних рішень, здатних компенсувати недостатність людського фактора [5].

Економічний аспект проблеми також має критичне значення. Високі витрати на професійний догляд роблять такі послуги недоступними для значної кількості населення, що потребує пошуку альтернативних підходів до забезпечення якісної підтримки певних груп користувачів. У цьому випадку поширення технології автономних роботів-помічників представляють собою перспективне рішення, що може забезпечити економічно ефективну альтернативу традиційним методам догляду.

Інтерактивні роботи-компаньйони репрезентують комплексні технологічні системи, що інтегрують передові розробки в галузі штучного інтелекту, сенсорних технологій та механічних маніпуляторів. Їх функціональність охоплює широкий спектр завдань від виконання рутинних побутових операцій до надання емоційної підтримки та здійснення терапевтичних процедур.

1.1 Аналіз конструкцій та апаратної архітектури інтерактивних роботів-компаньйонів

В умовах обмеженості людських ресурсів, у сфері догляду виникає потреба в розробці технологічних рішень, здатних частково або повністю замінити людський фактор у виконанні рутинних завдань. Саме роботи-компаньйони представляють собою комплексні технічні системи, що поєднують функції фізичної допомоги та емоційної підтримки, забезпечуючи комплексний підхід до покращення якості життя цільових груп користувачів.

Сучасні інтерактивні роботи-компаньйони являють собою комплексне поєднання передових технологій, включаючи системи, такі як, штучного інтелекту, сенсорні мережі, механізми та інтерфейси людино-машинної взаємодії.

Ці технологічні компоненти забезпечують можливість виконання широкого спектра функцій, які включають в себе від базових побутових операцій до складних терапевтичних процедур.

Сфери застосування таких систем охоплюють медичні заклади, установи соціального догляду, освітні інституції та приватні житлові приміщення.

Конструкційні особливості роботів-компаньйонів повинні враховувати специфічні вимоги щодо безпеки, ергономічності та функціональності. Прикладом технічно затребуваного рішення є робот для підтримки людини (Human Support Robot – HSR) [5], який поєднує мобільність та маніпулятивні можливості, мобільний маніпулятор здатний виконувати різноманітні фізичні завдання, включаючи піднімання об'єктів з різних поверхонь та положень.

Технічні характеристики HSR демонструють оптимальний баланс між функціональністю та безпекою використання.

Максимальна вантажопідйомність системи становить 1,2 кг в будь-якому положенні, що забезпечує можливість маніпулювання побутовими предметами

в трьох основних напрямках: вертикальному (згори), горизонтальному (збоку) та фронтальному (спереду).

Висота конструкції – 1,35 м – оптимізована для досягнення стандартних меблевих поверхонь на рівні плеча (1331 мм), що відповідає антропометричним характеристикам стоячої людини (рис. 1.1).

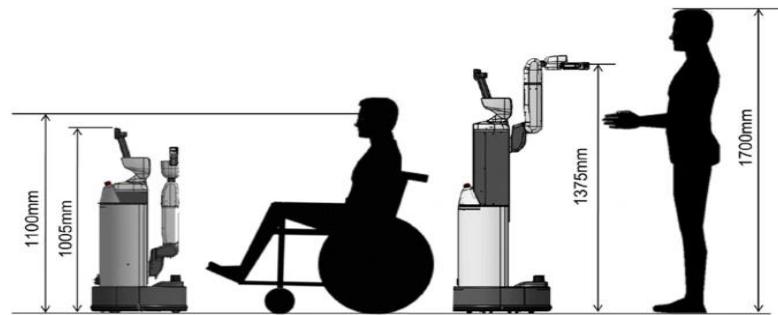


Рисунок 1.1 – Висота конструкції робота HSR [5]

Параметри швидкості пересування обмежені до 0,8 км/год, що забезпечує безпечну взаємодію з користувачами, особливо з представниками вразливих груп населення.

Ширина конструкції не перевищує 50 см, що дозволяє паралельне переміщення поруч з електричними інвалідними візками, відповідно до вимог міжнародних стандартів ISO-7176/5 та JIS-T 9203 (рис. 1.2).

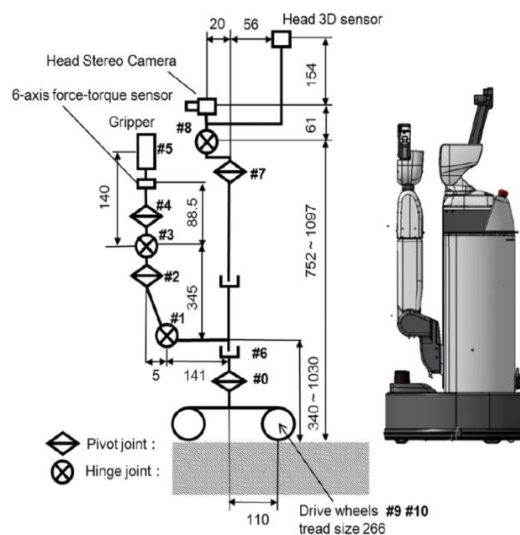


Рисунок 1.2 – Конструкторські характеристики робота HSR [5]

Ергономічні вимоги до конструкції включають обмеження висоти в стандартному положенні до рівня, що не перевищує 110 см (рівень очей користувача візка), що запобігає виникненню психологічного дискомфорту або почуття загрози у користувачів. Ключові компоненти робота помічника зображено на рис. 1.3.

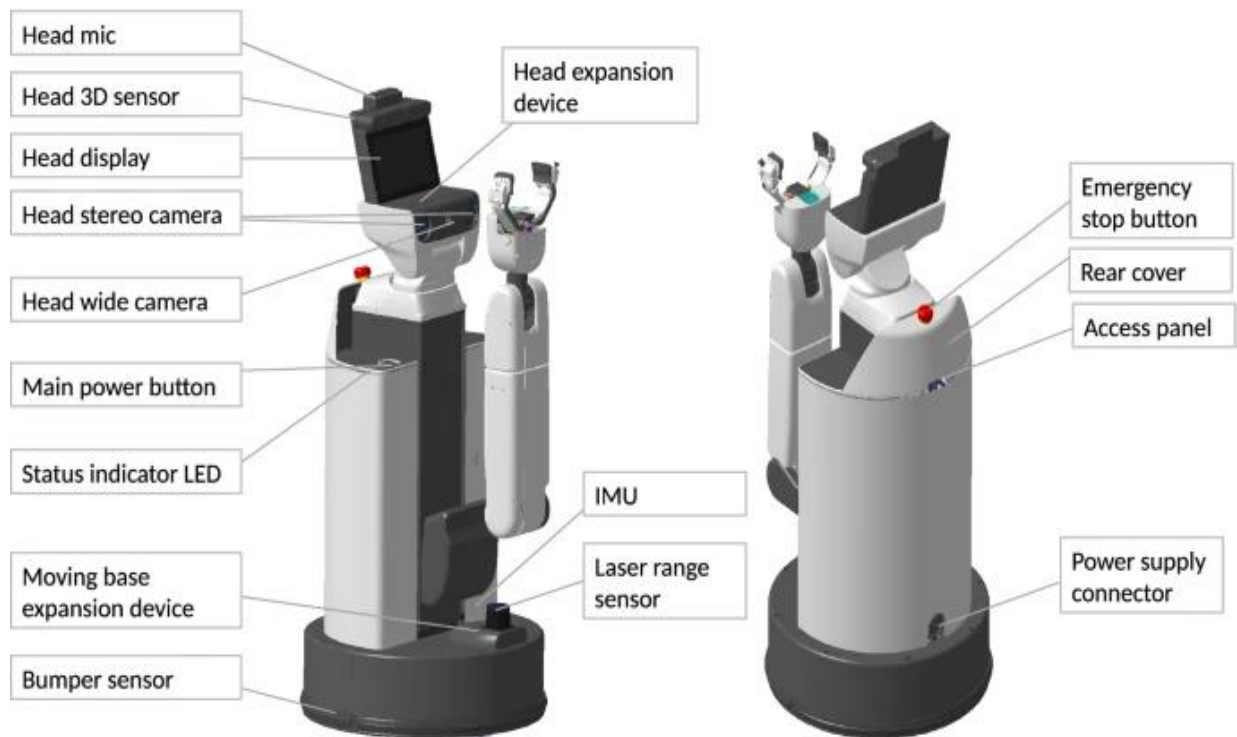


Рисунок 1.3 – Компоненти робота HSR [5]

Затребуваним альтернативним прикладом розвитку технологій роботів-компаньйонів є система Jibo [6] (рис. 1.4). Вона демонструє інтеграцію функціональних можливостей з емоційною підтримкою користувачів.

Характеризується високим рівнем інтерактивності, реалізованим через голосове управління та адаптивну поведінку.

Конструкція системи включає сенсорні компоненти та механізми обертання, що забезпечують інтуїтивну взаємодію з навколишнім середовищем.

Конструкційні рішення Jibo ґрунтуються на принципах безпечного дизайну з використанням м'яких обводів та ергономічних форм. Це мінімізує

ризика травматизму при взаємодії з різними категоріями користувачів, включаючи дітей та осіб похилого віку.

Компактні розміри та невелика вага системи забезпечують ефективну експлуатацію в обмежених просторах, таких як житлові приміщення або медичні установи.



Рисунок 1.4 – Робот Jibo [6]

Функціональність Jibo поєднує виконання практичних завдань з наданням емоційної підтримки, що є критично важливим аспектом для систем, призначених для роботи з особами з обмеженими можливостями. Здатність системи до інтерактивної комунікації, включаючи відповіді на запитання та виконання різноманітних завдань, робить її ефективним інструментом для підтримки психологічного благополуччя користувачів [6].

Таким чином, аналіз конструкційних та апаратних рішень сучасних роботів-компаньйонів демонструє комплексний підхід до оптимізації параметрів ваги, габаритів, безпеки та функціональності.

Ці технологічні рішення створюють основу для розвитку інноваційних роботизованих систем, здатних значно покращити якість життя користувачів через надання персоналізованої фізичної та емоційної підтримки.

1.2 Аналіз систем управління та взаємодії інтерактивних роботів-компаньйонів з користувачами

Системи управління інтерактивних роботів-компаньйонів представляють собою комплексні архітектури, що поєднують елементи автономного прийняття рішень, сенсорної інтеграції та адаптивної взаємодії з користувачами.

Ефективність функціонування таких систем залежить від здатності інтегрувати множинні потоки даних від різноманітних сенсорів, обробляти їх у реальному часі та генерувати адекватні відповіді на динамічні зміни в навколишньому середовищі.

Архітектура програмного забезпечення роботів-компаньйонів базується на принципах модульності та розподіленої обробки, що забезпечує можливість одночасного виконання множинних завдань. Компонентами такої архітектури є модуль планування траєкторії, схема керування маніпуляторами, інтерфейс людино-машинної взаємодії та підсистема забезпечення безпеки.

Інтеграція сенсорних систем включає використання компонентів, які дозволяють досягти високого рівня ситуаційної обізнаності та адаптивності поведінки роботів у складних динамічних середовищах: лідарів для просторової орієнтації, камер для розпізнавання об'єктів, тактильних сенсорів для контролю зусиль при маніпуляціях, акустичних систем для голосової взаємодії.

Критичним аспектом розробки є забезпечення балансу між автономністю функціонування та безпекою взаємодії з користувачами. Це досягається через імплементацію багаторівневих систем контролю, таких, як механізми запобігання зіткненням, алгоритми детекції аномальних ситуацій та процедури екстреної зупинки операцій.

1.2.1 Порівняльний аналіз сучасних роботів-компаньйонів

Для комплексного розуміння сучасного стану технологій доцільно провести порівняльний аналіз основних представників ринку роботів-компаньйонів (табл. 1.1). Роботи-компаньйони, зокрема такі як HSR, Pepper та Jibo, є прикладами інтеграції цих технологій у реальному світі.

Таблиця 1.1 – Порівняльні характеристики роботів-компаньйонів

Характеристика	HSR (Toyota) [5]	Pepper (SoftBank) [7]	Jibo [6]
Функціональність	Мобільний маніпулятор	Соціальний помічник	Сімейний компаньйон
Висота, м	1,35	1,20	0,28
Вантажопідйомність, кг	1,2	0,5	-
Швидкість, км/год	0,8	3,0	-
Тип взаємодії	Голосова/ тактильна	Голосова/ візуальна	Голосова/ візуальна
Автономність, год	2-3	10-12	4-6
Середовище застосування	Житлове	Комерційне/ публічне	Житлове

HSR здатний виконувати завдання по переміщенню предметів, управлінню побутовою технікою та наданню допомоги в прибиранні, що забезпечує значну підтримку для осіб з обмеженими фізичними можливостями (рис. 1.5).

Важливим досягненням у розробці таких роботів є здатність інтегрувати їх у реальні умови житлових приміщень, де вони можуть ефективно співіснувати з людьми, виконуючи функції помічників.

Завдяки використанню інтегрованих сенсорів та маніпуляторів, HSR може виконувати складні завдання, що включають переміщення предметів та взаємодію з побутовими пристроями, забезпечуючи комфорт і зручність для користувачів.



а)

б)



в)

а) сценарій 1;

б) сценарій 2;

в) сценарій 3

Рисунок 1.5 – Можливі сценарії взаємодії з HSR [5]

Рерреr від SoftBank Robotics використовує технології соціальної взаємодії з користувачем та був розроблений для тривалих соціальних взаємодій, здатний спілкуватися з користувачами через мову та жести (рис. 1.6).

Рерреr реагує на емоційні стани користувачів, демонструючи емпатію та відповідні реакції. Взаємодія з користувачем здійснюється через заздалегідь

задані фрази та реакції на основі емоційного контексту, що дозволяє роботам створювати більш людські та природні взаємодії.



Рисунок 1.6 – Робот Pepper [7]

Pepper взаємодіє з користувачем за допомогою попередньо записаних запитань і відповідей, контрольованих за допомогою техніки WoZ. Методика дозволяє експериментатору контролювати роботу, натискаючи відповідні клавіші для ініціювання запитів, що робить процес комунікації більш контрольованим і адаптованим до потреб користувачів. Завдяки високому голосу та анімованим рухам тіла, Pepper намагається створити природній вигляд взаємодії, що є важливим для соціальних роботів, які взаємодіють з людьми на емоційному рівні.

HSR і Pepper, демонструють ефективне вирішення проблеми підтримки людей похилого віку та осіб з обмеженими можливостями, надаючи їм можливість покращити свою незалежність у побуті. Вони не лише виконують фізичну роботу, але й активно взаємодіють з користувачами, забезпечуючи комфорт і безпеку в процесі своєї роботи.

Аналіз переваг та недоліків існуючих рішень показує, що HSR характеризується високим рівнем маніпуляційних можливостей, адаптованістю до житлових умов та відповідністю міжнародним стандартам безпеки, проте

має обмежену автономність, високу вартість та складність обслуговування. Реррег відрізняється тривалою автономною роботою, розвиненими соціальними функціями та комерційною доступністю, але має обмежені фізичні можливості та залежність від попередньо запрограмованих сценаріїв.

Jibo демонструє компактність, інтуїтивний інтерфейс та емоційну адаптивність, однак характеризується відсутністю мобільності, обмеженою функціональністю.

1.2.2 Системи прийняття рішень у роботах-компаньйонах

Прийняття рішень в автономних роботах-компаньйонах є багатокомпонентним процесом, що включає збір та аналіз сенсорної інформації, оцінку поточного стану середовища, прогнозування наслідків можливих дій та вибір оптимальної стратегії поведінки. Архітектура систем прийняття рішень включає рівень сприйняття (Perception Layer), що відповідає за обробку сенсорних даних та формування моделі навколишнього середовища, рівень планування (Planning Layer), що генерує можливі сценарії дій та оцінює їх доцільність, та рівень виконання (Execution Layer), що реалізує обрані дії та здійснює моніторинг результатів [8] (рис. 1.7).

Алгоритмічні підходи включають використання марковських процесів прийняття рішень (MDP) для моделювання ситуацій з невизначеністю, де робот має оцінити ймовірні наслідки своїх дій. Формально MDP описується кортежем (S, A, P, R), де:

- S – є множиною станів;
- A – є множиною дій;
- P – являє собою матрицю ймовірностей переходів;
- R – є функцією винагороди.

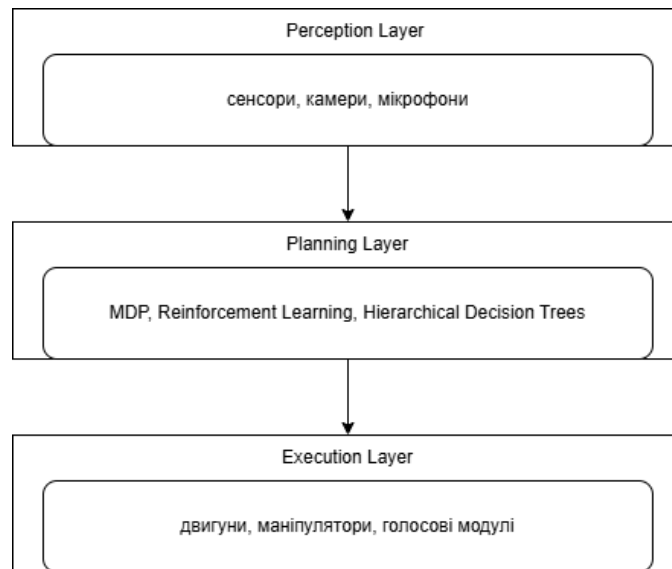


Рисунок 1.7 – Архітектура системи прийняття рішень

Підкріплювальне навчання (Reinforcement Learning) дозволяє роботам адаптувати свою поведінку через взаємодію з середовищем, де алгоритми Q-learning та Deep Q-Networks (DQN) забезпечують можливість навчання оптимальних стратегій без попереднього знання моделі середовища.

Ієрархічні дерева рішень структурують процес прийняття рішень у вигляді деревоподібної архітектури, де кожен вузол представляє точку прийняття рішення, а гілки відповідають можливим варіантом дій.

Практичне застосування цих підходів демонструють сучасні системи роботів-компаньйонів.

У системі HSR використовується гібридний підхід, що поєднує детерміновані алгоритми для базових операцій (навігація, захват об'єктів) з адаптивними механізмами для складних сценаріїв взаємодії з користувачами [6].

Perper застосовує емоційно-адаптивні алгоритми, що аналізують вираз обличчя, тон голосу та жести користувача для вибору відповідної стратегії комунікації, використовуючи нейронні мережі для класифікації емоційних станів та генерації відповідних реакцій.

Якщо ж використовує комбіновану систему прийняття рішень, яка включає в себе ієрархічні дерева рішень, суспільно-орієнтовану поведінку моделі, а також алгоритми машинного навчання.

Сучасні розробки включають інтеграцію великих мовних моделей (LLM) [8] для покращення природності діалогу та розширення можливостей розуміння контексту взаємодії, що дозволяє роботам генерувати більш релевантні та персоналізовані відповіді на запити користувачів.

Основними викликами залишаються забезпечення прозорості процесу прийняття рішень, гарантування безпеки в непередбачуваних ситуаціях та адаптація до індивідуальних потреб користувачів.

Перспективними напрямками розвитку є інтеграція мультимодальних інтерфейсів, запровадження федеративного навчання та розробка етичних фреймворків для автономних рішень.

Роботи HSR і Pepper демонструють різні підходи до вирішення завдань підтримки людей похилого віку та осіб з обмеженими можливостями, де HSR орієнтований на фізичну допомогу через маніпуляційні операції, а Pepper спеціалізується на соціальній взаємодії та емоційній підтримці, обидва підходи сприяють підвищенню рівня незалежності користувачів та покращенню якості їх життя.

1.3 Аналіз технологій адаптивної підтримки та персоналізації для людей з обмеженими можливостями

Забезпечення доступності програмного забезпечення для користувачів з обмеженими можливостями є критично важливою задачею сучасної інформатики. Створення адаптивних програмних рішень, здатних враховувати індивідуальні потреби користувачів, становить особливий інтерес для дослідників у галузі людино-машинної взаємодії. Розробка мобільних додатків

з адаптивними інтерфейсами та функціональністю може суттєво покращити якість життя людей з різними формами обмежень.

Одним із найперспективніших підходів до вирішення даної проблеми є інтеграція онтологічних моделей у програмні системи (рис. 1.8). Онтологічне моделювання дозволяє формалізувати знання про предметну область та забезпечити адаптацію функціональності програми до динамічних змін у вимогах або середовищі функціонування. Онтологічні структури формують концептуальну основу для підвищення персоналізації та гнучкості програмного забезпечення, забезпечуючи індивідуалізований підхід до кожного користувача.

Основною перевагою онтологічного моделювання є його здатність адаптувати програмні системи до нових вимог без необхідності кардинального перепроектування архітектури та переписування основного програмного коду, забезпечуючи підвищену гнучкість системи та знижує часові й ресурсні витрати, необхідні для розробки нових версій продукту [9].

Онтологічний підхід дозволяє ефективно інтегрувати експертні знання у конкретній предметній області без необхідності залучення програмістів до кожної модифікації системи.

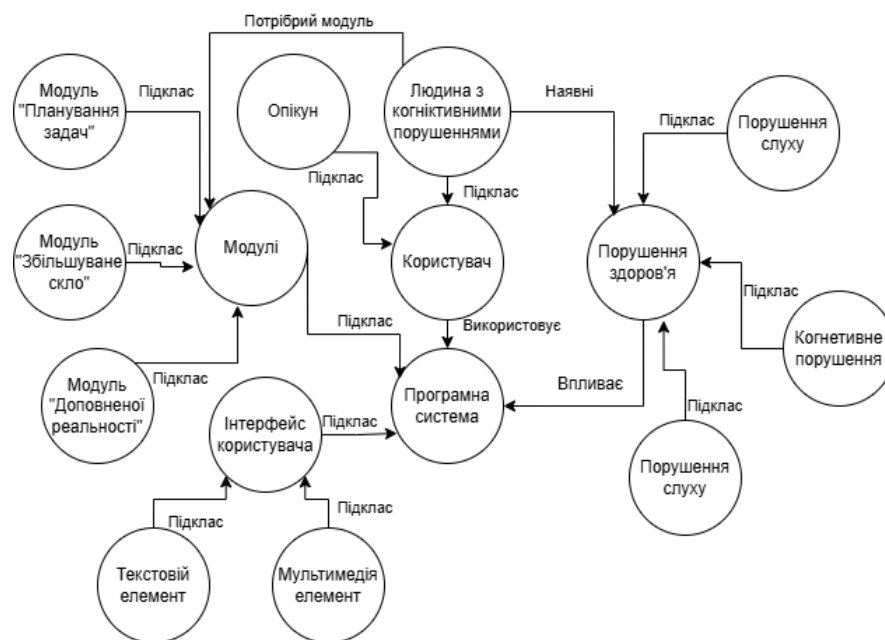


Рисунок 1.8 – Зображення онтологічної моделі адаптивної програмної системи

Онтологічні моделі можуть бути застосовані для побудови адаптивних інтерфейсів, які автоматично налаштовують користувацький інтерфейс та функціональність відповідно до специфічних потреб користувача. Наприклад, для користувачів з когнітивними порушеннями можна реалізувати систему яка динамічно адаптує графічні елементи інтерфейсу, враховуючи індивідуальні вимоги, такі як підвищена контрастність або підтримка зорового сприйняття через збільшувальні елементи [9].

Використання адаптивних моделей сприяє, зниженню кількості помилок при взаємодії з системою та покращенню користувацького досвіду осіб з обмеженими можливостями. Додатково забезпечується безперервне оновлення та персоналізація системи через постійну синхронізацію з онтологічною базою знань, що дозволяє системі динамічно адаптуватися до нових умов або змін у стані користувача.

Дані моделі представляють собою ефективний інструмент для створення адаптивних програмних систем, які можуть суттєво покращити користувацький досвід людей з обмеженими можливостями, забезпечуючи персоналізовані рішення та адаптуючи програму до конкретних потреб кожного користувача.

1.3.1 Прийняття рішень у системах автономних мобільних роботів

Прийняття рішень – є фундаментальним компонентом архітектури автономних мобільних роботів, що забезпечує адаптацію їхньої поведінки, відповідно, до динамічних змін у навколишньому середовищі та специфічних вимог користувачів.

Процес складається зі збору та обробки сенсорних даних, оцінки поточної ситуації та вибору оптимальних дій для досягнення поставлених цілей.

Ключовим аспектом таких систем є здатність робота до автономного прийняття рішень без необхідності безперервного втручання людини-оператора.

У сучасній робототехніці для досягнення автономності використовуються різноманітні алгоритмічні підходи та математичні моделі (марковські процеси прийняття рішень (MDP) (рис. 1.9), алгоритми підкріплювального навчання (Reinforcement Learning), дерева рішень).



Рисунок 1.9 – Марковський процес прийняття рішень [10]

Марковські процеси прийняття рішень дозволяють роботам оцінювати ймовірні наслідки своїх дій і формувати стратегії, які максимізують очікувану винагороду. Це особливо актуально для мобільних роботів, що функціонують у складних та динамічних середовищах. Зокрема, ці алгоритми дозволяють оптимізувати траєкторії руху та реалізувати ефективні стратегії уникнення перешкод [10].

Алгоритми підкріплювального навчання (RL) забезпечують роботам можливість навчатися на основі взаємодії з середовищем, отримуючи винагороди чи покарання за виконані дії. Це дозволяє роботам адаптуватися до нових умов без необхідності перепрограмування базової архітектури.

Дерева рішень, у свою чергу, моделюють можливі варіанти дій та їхні наслідки, забезпечуючи роботам можливість прийняття обґрунтованих рішень у складних ситуаціях, де вибір може залежати від множини факторів.

Практичним прикладом застосування адаптивних алгоритмів прийняття рішень є система, що використовується у роботизованих комплексах для логістики, зокрема у розподільних центрах Amazon. Роботи застосовують складні алгоритми для транспортування пакунків, сортування товарів та створення індивідуальної упаковки, що дозволяє адаптувати поведінку роботів залежно від умов середовища, підвищуючи операційну ефективність та зменшуючи ймовірність помилок.

Іншим показовим прикладом є гуманоїдний робот Pepper, який використовує адаптивні алгоритми для взаємодії з користувачами, налаштовуючи свою поведінку залежно від емоційних та фізичних реакцій людини. Робот активно застосовується для надання допомоги людям з обмеженими можливостями, адаптуючи свої дії відповідно до потреб користувача, наприклад, модифікує голосові характеристики або поведінкові патерни для створення більш комфортного середовища взаємодії [7].

Значний інтерес представляє розвиток автономних безпілотних літальних апаратів (БПЛА), здатних самостійно приймати рішення в реальному часі. Наприклад, рій автономних дронів, розроблений угорськими дослідниками, самостійно обирає оптимальні траєкторії руху, уникаючи зіткнень, та адаптується до нових умов середовища. Дані системи забезпечують автономність та ефективність без централізованого контролю, що дозволяє дронам функціонувати в складних умовах, таких як нічний час або умови обмеженої видимості.

Застосування адаптивних алгоритмів для прийняття рішень дозволяє мобільним роботам досягати вищої ефективності та автономності, що сприяє розвитку роботизованих систем у різних сферах застосування, від логістики до надання допомоги людям з обмеженими можливостями, технології перебувають у стані постійного вдосконалення, що дозволяє роботам ставати більш гнучкими та здатними до адаптації в різноманітних операційних середовищах.

2 РОЗРОБКА МАКЕТУ РОБОТА-ПОМІЧНИКА

2.1 Розробка загальної структурної схеми робота-помічника

Перед початком конструкторської діяльності, було розроблено структурну схему системи управління мобільного робота-помічника.

На рис. 2.1 зображено взаємодію основних апаратних та програмних модулів об'єкта розробки, які виконують основні функції взаємодії з навколишнім середовищем, а саме орієнтацію в просторі, комунікацію з користувачем, а також автономну навігацію.

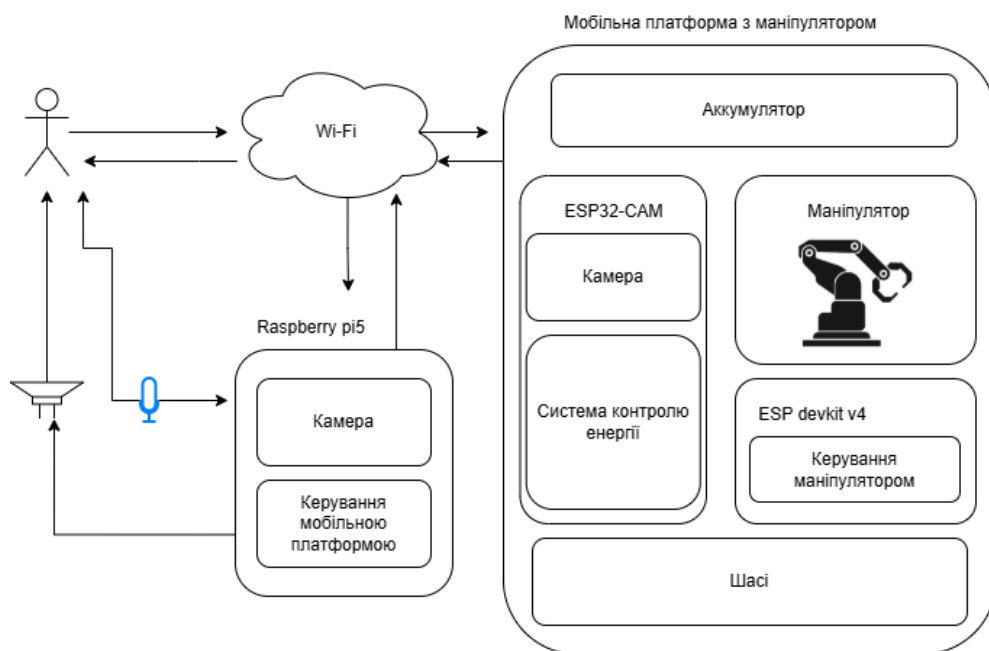


Рисунок 2.1 – Структурна схема голосового помічника та мобільного робота-помічника

Структурна схема мобільного робота-помічника включає в себе комплекс апаратних компонентів таких як, головна плата керування та мобільна платформа з маніпулятором.

Важливу роль у функціонуванні приладів відіграє HTTP протокол передачі даних. Завдяки тому, що плати містять в собі Wi-Fi модуль, віддалена

комунікація здійснюється за REST API. Це дозволяє працювати з віддаленими API для нарощування функціоналу та додавання в систему приладів існуючі програмні рішення щоб, забезпечити краще функціонування.

В якості джерела живлення мобільного робота з маніпулятором виступає акумуляторний блок розміщений на платформі для забезпечення автономної роботи на певний час. Робот-помічник живиться через дротове підключення, так як в даній реалізації він буде залишатися на статичному місці.

Мобільний робот з маніпулятором розташований на металевому корпусі з шасі, яке виконує функції несучої платформи, що містить на собі електроприводи та колісну базу. Конструкція шасі забезпечує маневреність пристрою за допомогою сенсорів, які знаходяться на чотирьох сторонах для обробки відстані до перешкоди. Це дозволяє програмно обробляти випадки, коли робот наштовхнеться чи буде близько до певного об'єкта.

Система керування рухом виконує обробку команд, отриманих від головного контролера, реалізує рух мобільної платформи. Разом з запитом від керуючої плати, обробляє дані від ультразвукових датчиків відстані, за допомогою яких і реалізований алгоритм обходу перешкод або зупинка й припинення руху в певну сторону для запобігання наїзду. Отримуючи зображення з камери на платформі, можна також скоригувати маршрут та досягти більш оптимального рішення.

Для підвищення інтерактивності та персоналізації робота з користувачем передбачено використання камери контролю емоційного стану, включаючи стать та вік, для отримання більшої кількості інформації з зображення. Камера інтегрована у систему та застосовує методи комп'ютерного зору для аналізу міміки. Таким чином, відбувається адаптація відповіді пристрою, відповідно до психоемоційного стану людини.

Для розширення функціональних можливостей мобільного робота, було прийнято рішення про доповнення маніпулятором. Він переміщує, захоплює та виконує маніпуляції певними об'єктами, за потребами користувача, що значно

покращує спектр можливих сфер застосувань робота. Така механічна рука, керується окремим обчислювальним модулем ESP Devkit V4.

Для забезпечення навігації мобільного робота, було обрано ESP32-CAM, яка приймає на засіб введення інформації зображення навколишнього середовища для подальшого аналізу перешкод чи моделювання моделі поведінки робота. Дане технічне рішення надає можливість орієнтуватися у просторі, бачити користувачеві оточення, в разі знаходження робота в іншому приміщенні.

Центральною керуючою платою системи є головний контролер на базі мікрокомп'ютера Raspberry Pi 5. До неї підключено камеру, мікрофон та колонки. Плата виконує обробку вхідних даних від камери та голосового модуля, на який надходить запит від користувача. Контролер відповідає за формування керуючих команд, інтеграцію алгоритмів штучного інтелекту та забезпечення безперебійної взаємодії між всіма функціональними модулями робота.

Розроблена структурна схема мобільного робота з маніпулятором та голосовим помічником демонструє взаємозв'язок внутрішніх модулів та спілкування керуючих плат.

Інтеграція апаратних і програмних компонентів дозволяє забезпечити функціонування пристрою та має можливість, в подальшому, приєднувати додаткові модулі для розширення функціоналу.

2.2 Вибір апаратних модулів

На першому кроці було проведено вибір та обґрунтування компонентів робота-компаньйона для людей з обмеженими можливостями, було обрано компоненти і проаналізовано їх сумісність.

Наступним кроком, було обрано головну плату, а саме одноплатний комп'ютер Raspberry Pi 5. Через свої характеристики та потужність для роботи

зі штучним інтелектом, обробкою зображення та координацією всієї системи в цілому Raspberry Pi 5. було достатньо. Створення однієї головної точки входу надавало користувачу змогу швидко керувати всіма підключеннями.

На рис. 2.2 наведено одноплатний комп'ютер Raspberry Pi 5.



Рисунок 2.2 – Raspberry Pi 5 [11]

Головними критеріями вибору стали компактність конструкції (90×60 мм), наявність вбудованої підтримки бездротових мереж (Wi-Fi 5, Bluetooth 5.0) та широка екосистема програмного забезпечення. Чотирьохядерний процесор ARM Cortex-A76 з тактовою частотою 2.4 ГГц та обсяг оперативної пам'яті до 4 ГБ LPDDR4 забезпечують достатню продуктивність для обробки відеопотоку, розпізнавання мови та виконання навігаційних алгоритмів у режимі реального часу.

До мобільної платформи було додано плату ESP32-CAM, в якості компонента обробки зображення та надсилання відео потоку на керуючу плату. Це надає можливість відслідковувати навколишній простір та керувати роботом, знаходячись в іншому приміщенні, віддалено, дистанційно.

На рис. 2.3 наведено зображення камерного модуля ESP32-CAM.



Рисунок 2.3 – ESP32-CAM [12]

Модуль поєднує двоядерний мікроконтролер ESP32 з тактовою частотою 240 МГц, 2-мегапіксельну камеру OV2640 та підтримку бездротових протоколів Wi-Fi і Bluetooth у компактному корпусі розмірами 27×40 мм. Споживання модулем струму на рівні 180 мА забезпечує тривалу автономну роботу системи, що є критично важливим для мобільних роботизованих платформ.

Для одноплатного комп'ютера Raspberry Pi 5 було обрано ширококутну камеру (через гарну сумісність) та додатково придбаний дріт (для з'єднання з платою), щоб продовжити відстань та забезпечити можливість налаштування камери в положення зручне для користувача.

На рис. 2.4 наведено зображення камери для плати керування.



Рисунок 2.4 – Ширококутна камера для Raspberry Pi 5MP 1080P з ІЧ підсвічуванням [13]

Ширококутна камера має роздільну здатність 5 МР та підтримку запису відео у форматі 1080P. Кут огляду 160° дозволяє охопити значну область простору, що підвищує ефективність алгоритмів розпізнавання об'єктів та навігації.

Вбудоване інфрачервоне підсвічування забезпечує функціонування системи технічного зору в умовах недостатнього освітлення, розширюючи діапазон застосування робота.

В якості плати для керування рухом платформи та маніпулятором, обрано ESP32 DevKit V4.

Плата є однією з найпопулярніших технологічних рішень на базі Arduino для розробки роботів з великою кількістю підключення додаткових компонентів. Завдяки потужному двоядерному процесору Tensilica Xtensa LX6 з тактовою частотою до 240 МГц, є можливість одночасно обробляти задачі керування шасі та маніпулятором без значних затримок. Підтримка модулів Bluetooth та Wi-Fi дозволяє забезпечити зв'язок керуючих компонентів.

На рис. 2.5 наведено зображення модулю ESP32 DevKit V4.

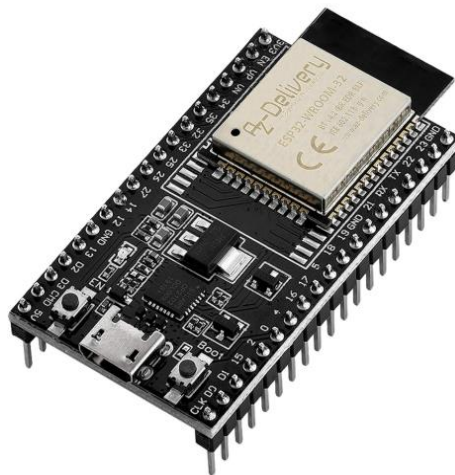


Рисунок 2.5 – Плата для розробки вбудованих систем ESP32 DevKit V4 [14]

Розробка системи керування маніпулятора для автономного мобільного робота передбачає використання серводвигунів для здійснення точних рухів, необхідних для захоплення об'єктів. У цьому випадку важливо вибрати

серводвигун, який здатний забезпечити потрібний обертаючий момент, енергоефективність та зручність інтеграції у конструкцію маніпулятора (рис. 2.6).



Рисунок 2.6 – Серводвигун SG90 [15]

Для реалізації рухів маніпулятора було обрано серводвигуни SG90, які забезпечують обертаючий момент 2 кг/см при споживанні струму 5 мА. Широкий діапазон робочої напруги (3.5-5 В) забезпечує сумісність з різними джерелами живлення та спрощує інтеграцію в загальну систему електроживлення робота. Габаритні розміри 33×30×13 мм дозволяють створити компактну конструкцію маніпулятора з достатньою механічною міцністю.

Для реалізації системи уникнення перешкод та автономної навігації було використано ультразвукові датчики HC-SR04, які забезпечують вимірювання відстані в діапазоні від 2 до 400 см, з точністю ± 3 мм.

На рис. 2.7 наведено зображення датчика відстані HC-SR04.



Рисунок 2.7 – Датчик відстані HC-SR04 [16]

Простота інтерфейсу підключення, низьке енергоспоживання (15 мА) та висока надійність роботи в температурному діапазоні від -20°C до 70°C роблять датчики оптимальним вибором для мобільних роботизованих систем. Широка підтримка програмних бібліотек та наявність великої кількості документації спрощують процес інтеграції та налагодження системи.

Обрані апаратні компоненти забезпечують реалізацію всіх запланованих функцій мобільного робота з маніпулятором та голосового помічника при мінімальному енергоспоживанні та максимальній надійності роботи системи. Модульна архітектура дозволяє здійснювати подальше вдосконалення окремих підсистем без кардинальних змін у загальній конструкції робота.

2.3 Розробка схеми підключення

При створенні даного робота було розроблено схему підключення мобільного робота з роботизованою рукою на основі мікроконтролера ESP32 DevKit. Даний робот є допоміжним механізмом основної керуючої системи. Він допомагає користувачеві здійснювати певну кількість віддалених дій, таких як захоплення об'єктів та їх транспортування, в разі відсутності можливості виконання даних дій самостійно.

Розробка почалась з того, щоб створити платформу, що рухається. Для цього було обрано мостовий драйвер L298N, до якого було поєднано два двигуна постійного струму. Було прийнято рішення підключити до живлення 5 V для запуску механізму руху. З'єднання з керуючою платою, в даному випадку виглядає так: мікроконтролер ESP32 DevKit передає двигунам сигнали Low через GPIO19 та GPIO18 до in1 та in2, High відповідно до GPIO21 та GPIO22 до in3 та in4.

Наступним елементом задачі було додавання серводвигунів для створення механічної руки. Для цього з'єднано кожний серводвигун до плати за перший – GPIO32, другий – GPIO25, третій – GPIO26, четвертий – GPIO27.

Кожен із серводвигунів має три виводи: сигнальний, живлення VCC та заземлення. Сигнальні контакти були з'єднані з вказаними GPIO ESP32. Живлення подано від системного джерела. А заземлення усіх елементів було об'єднано з загальним GND для коректної роботи.

Останніми елементами робота, що були додано, це ультразвукові датчики HC-SR04. Їх було додано для запобігання зіткнення з перешкодами на шляху та створення можливості керувати маршрутом робота. Передбачено встановлення чотирьох одиниць по чотирьох сторонах відповідно. Кожний з датчиків працює за принципом відправки ультразвукового імпульсу та вимірювання часу, за який сигнал повертається після відбиття від об'єкта.

З платою вони з'єднані через GPIO4 – Trig та GPIO2 – Echo, GPIO16 – Trig та GPIO35 – Echo, GPIO17 – Trig та GPIO36 – Echo, GPIO32 – Trig та GPIO39 – Echo. Завдяки тому, що GPIO32, GPIO25, GPIO26, GPIO27 універсальні виводи, які дозволяють генерувати стабільний PWM-сигнал, їх було обрано для з'єднання з Trig. Виводи GPIO2, GPIO35, GPIO36, GPIO39 використані як Echo, тому що це вхідні сигнали, куди повертається відбитий імпульс.

Для отримання зображення оточення та розуміння навколишнього середовища, розташування об'єктів в приміщенні для захоплення маніпулятором, було прийнято рішення про додавання камери, в якості якої було обрано плату ESP32. На ній створюється веб сервер, до якого підключається Raspberry Pi для виводу юзеру зображення.

На схемі також винесено живлення та заземлення в єдиний ланцюг, для запобігання сплутування та більш легкого розташування елементів на платформі.

Розроблена схема підключення враховує всі необхідні компоненти та їх взаємозв'язок (рис. 2.8).

Завдяки додавання ультразвукових датчиків відстані було додано важливий функціонал для маневрування робота. Камера доповнює картинку навколишнього простору користувачу.

2.4 Розрахунок системи управління

Для моделювання поведінки, було здійснено розрахунки передавальної функції серводвигуна на основі заданих параметрів [17]. Для розрахунків були використані наступні значення:

- опір обмотки ($R = 3$) Ом;
- константа крутного моменту ($k_t = 0.015$) Н·м/А;
- константа проти-ЕРС ($k_b = 0.015$) В·с/рад;
- момент інерції ротора ($J = 0.015$) кг·м²;
- коефіцієнт в'язкого тертя ($b = 0.05$) Н·м·с/рад.

Передавальна функція $G(s)$ серводвигуна була отримана за допомогою рівняння (2.1):

$$G(s) = \frac{k_t}{s(Js + b)(R + k_b s)}, \quad (2.1)$$

Підставивши значення параметрів, отримано спрощену передавальну функцію (2.2):

$$G(s) = \frac{0.015}{0.000225s^2 + 0.795s + 0.15}, \quad (2.2)$$

Передавальна функція описує динаміку серводвигуна та дозволяє здійснити подальший аналіз його поведінки.

Для розрахунку і аналізу системи був використаний MATLAB. Задано чисельник та знаменник передавальної функції, потім створено об'єкт системи

та виконано декілька важливих характеристик, таких як Найквіста та поступова характеристики, для дослідження її реакції.

```
numerator = [0.015];
denominator = [0.000225, 0.795, 0.15];
```

```
% Створення передавальної функції
```

```
W = tf(numerator, denominator);
```

```
% Аналіз
```

```
step(W)    % Поступова характеристика
```

На рис. 2.9 зображено поступову характеристику для серводвигунів.

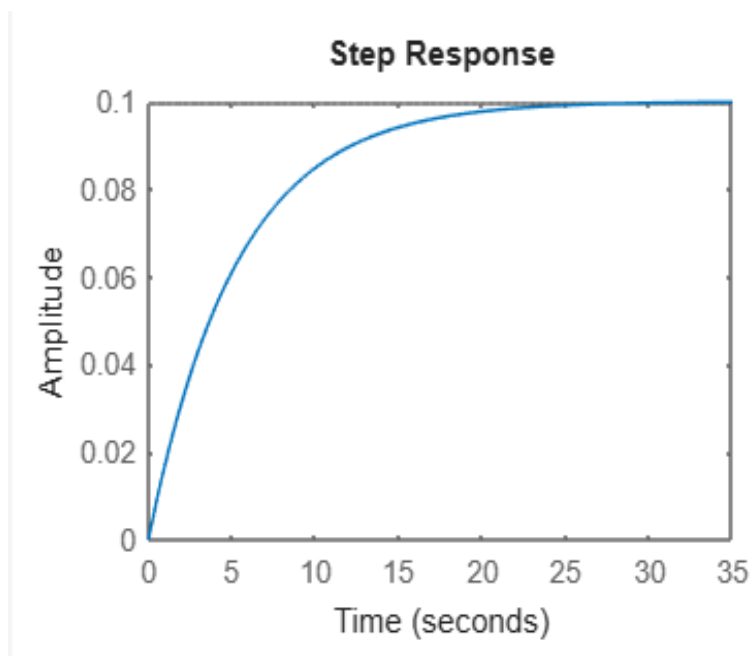


Рисунок 2.9 – Поступова характеристика

```
pole(W)    % Полюси системи
```

На рис. 2.10 зображено полюси системи для серводвигунів.

```
>> pole(W)

ans =

    1.0e+03 *
   -3.5331
   -0.0002
```

Рисунок 2.10 – Полюси системи

```
pzmap(W) % Полюси та нулі
```

На рис. 2.11 зображено полюси та нулі системи.

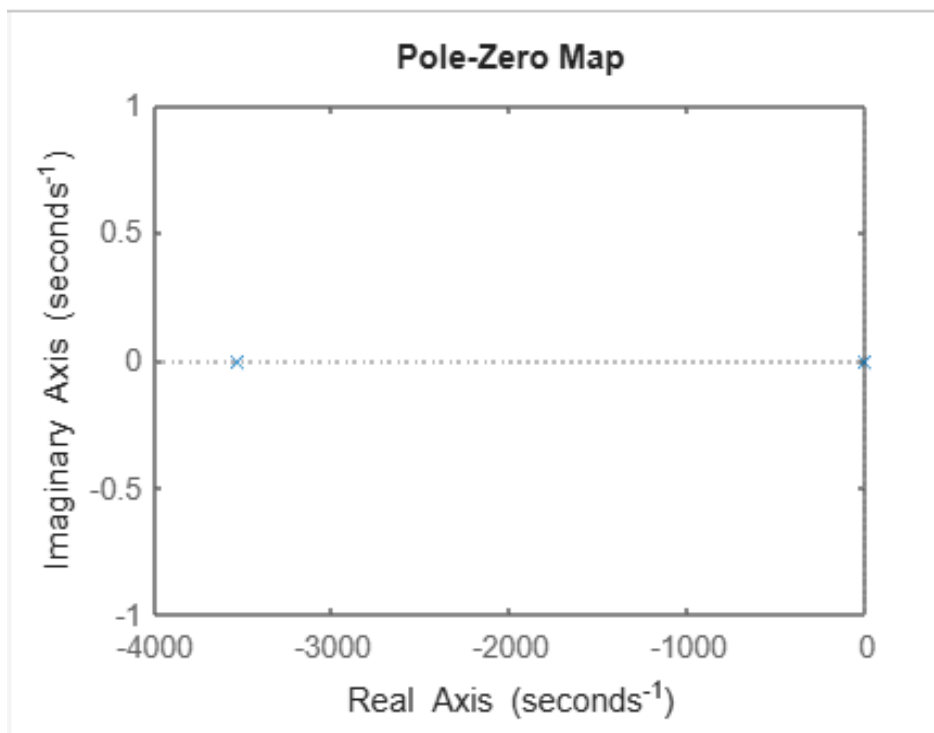


Рисунок 2.11 – Полюси та нулі

```
nyquist(W) % Характеристика Найквіста
```

На рис. 2.12 зображено Найквіста характеристика системи.

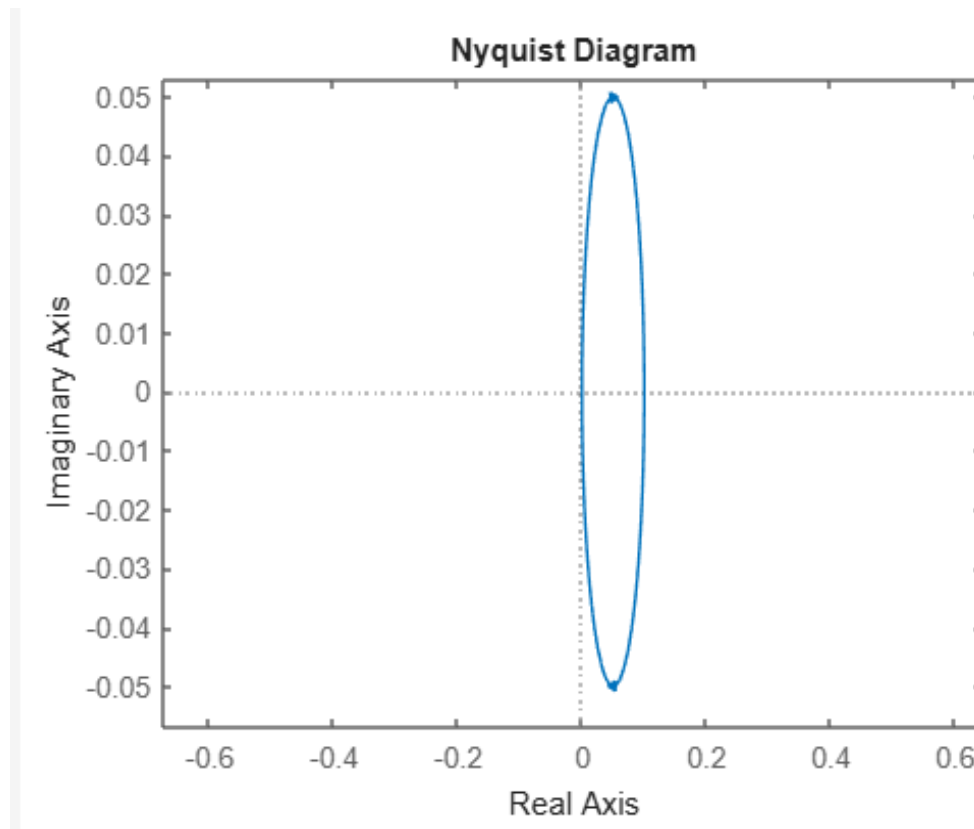


Рисунок 2.12 – Графік характеристики Найквіста

У результаті аналізу отримано кілька ключових характеристик системи. Поступова характеристика показує, як система реагує на одиничний ступінь вхідного сигналу, що дозволяє оцінити час встановлення, перерегулювання та стабільність системи. Аналіз полюсів системи вказує на її стійкість, бо якщо полюси мають від'ємні дійсні частини, система вважається стійкою.

Графік полюсів та нулів дає можливість візуалізувати місце розташування критичних точок у комплексній площині, що є важливим для аналізу стабільності та динамічних характеристик системи.

Характеристика Найквіста дозволяє оцінити частоту поведінку системи та її стійкість на різних частотах вхідного сигналу.

Система стабільна і здатна ефективно реагувати на зміни вхідного сигналу без виникнення небажаних коливань або нестабільностей.

2.5 Моделювання робота в середовищі Tinkercad

Для розробки та тестування мобільної платформи з механічною рукою було обрано середовище Tinkercad, яке дозволяє швидко моделювати електронні схеми, імітувати реальну поведінку в ідеальному світі, програмувати мікроконтролери та проводити віртуальні експерименти без необхідності використовувати фізичні компоненти [18].

Вибір Tinkercad зумовлений його простотою, доступністю та інтеграцією з платформою Arduino, що дозволяє ефективно тестувати логіку керування роботом.

Основним контролером – Arduino Uno, який було обрано для імітування поведінки ESP32 Dev Kit, обробляє команди та керує всіма компонентами.

Для забезпечення руху використано два електродвигуни, швидкість і напрямок яких регулюються через драйвер L293D. Драйвер дозволяє змінювати обертання моторів, що забезпечує можливість керованого пересування робота.

Маніпулятор керується за допомогою чотирьох сервоприводів SG90, які відповідають за рух його окремих частин. Завдяки цим сервоприводам, робот може здійснювати маніпуляції з об'єктами в межах досяжності.

Для аналізу навколишнього середовища та запобігання зіткненням у конструкції використано ультразвукові датчики HC-SR04. Вони вимірюють відстань до об'єктів та передають отримані дані на Arduino Uno, що дає можливість автоматично коригувати маршрут руху робота. Ультразвукові датчики постійно аналізують середовище, що дає змогу роботу змінювати напрямок руху при виявленні перешкод.

Змодельована схема (рис. 2.13) імітує основні функції мобільної платформи, яка може переміщуватися в просторі та керувати маніпулятором.

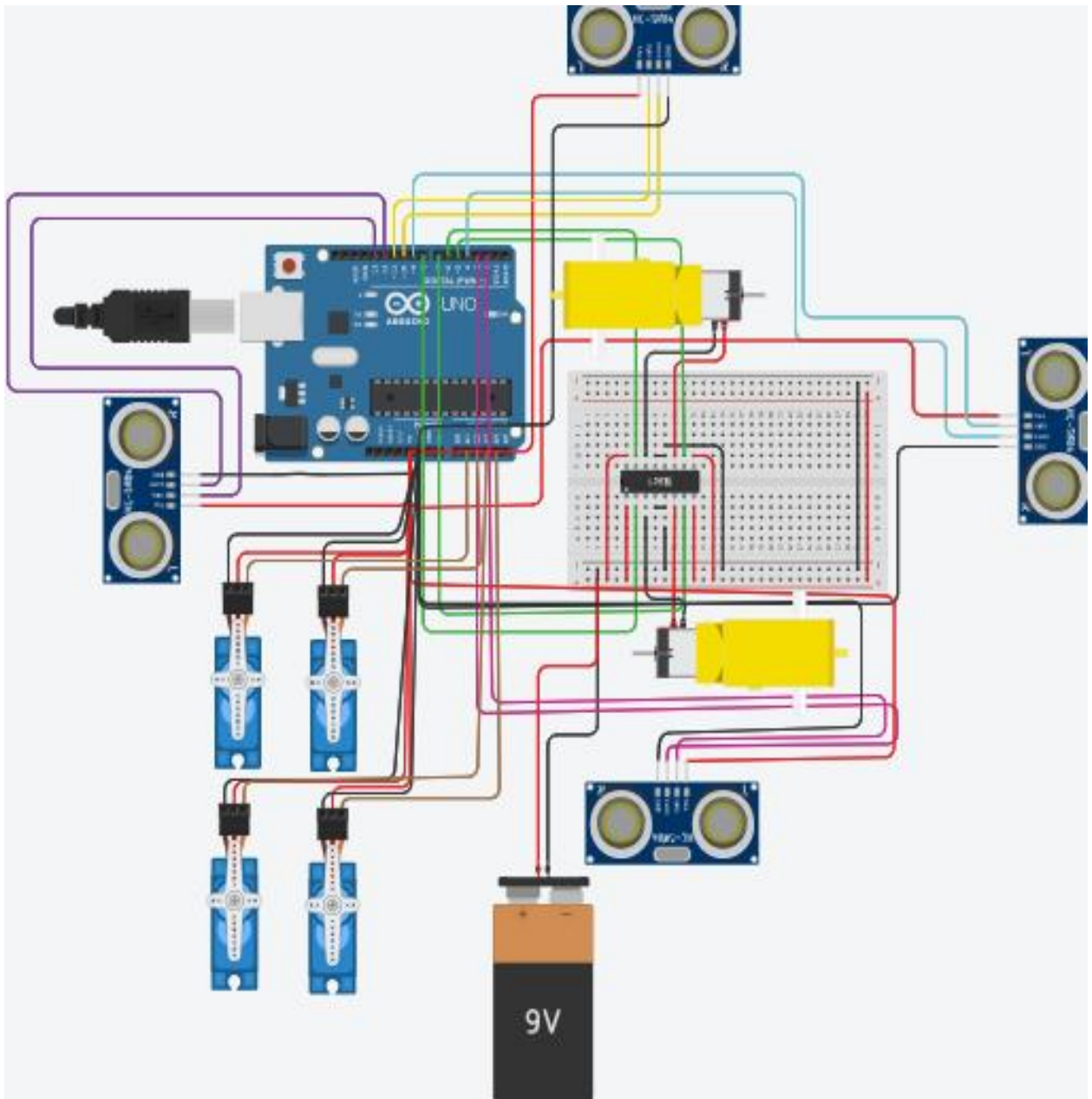


Рисунок 2.13 – Змодельована схема

Побудова моделі дала змогу провести імітаційне моделювання поведінки робота, відлагодити код програми мобільної платформи та механічної руки. Це значно спростило процес розробки та налагодження роботи програмного забезпечення. Дало можливість легко виявити помилки та виправити їх не використовуючи реальні компоненти.

2.6 Збірка макету робота

Процес збирання макету робота розпочався з підготовки конструктивних елементів, необхідних для забезпечення надійного кріплення ультразвукових датчиків відстані, маніпулятора, двигунів та інших компонентів. Основним матеріалом для корпусу та кріпильних елементів було обрано пластик завдяки його легкості, міцності та зручності в обробці.

Першим етапом було виготовлення пластикових подовжувачів, призначених для кріплення ультразвукових датчиків HC-SR04 (рис. 2.14) на платформі. Подовжувачі виконують функцію регулювання розташування датчиків, дозволяючи налаштувати їх таким чином, щоб вони могли ефективно визначати перешкоди перед роботом. Для надійної фіксації використовувалися болтові з'єднання та додаткові фіксатори.

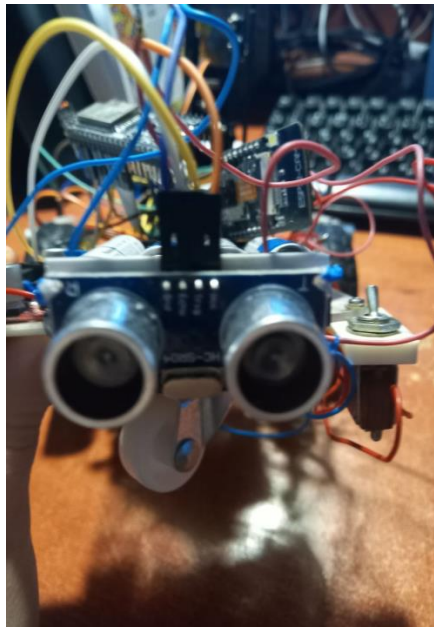


Рисунок 2.14 – Розміщення сенсорів

На основну пластину також були нанесені маркування для точного розміщення маніпулятора. Вона була виготовлена з урахуванням навантажень,

які виникають при роботі сервоприводів, що дозволяє мінімізувати деформації та забезпечити стабільність конструкції.

Після визначення оптимального розташування подовжувачів для датчиків, наступним етапом було встановлення двигунів та колісної бази.

Робот оснащений двома незалежними мотор-редукторами, що дозволяють здійснювати рух у різних напрямках. Для управління двигунами було використано драйвер L298N, який дозволяє керувати напрямком обертання та швидкістю обертання коліс за допомогою ШІМ-сигналів.

Перед закріпленням двигунів було перевірено їх працездатність, зокрема, протестовано відгук на керуючі сигнали та визначено оптимальні параметри ШІМ для забезпечення плавності руху. Особливу увагу було приділено розташуванню коліс, щоб уникнути механічного контакту між ними під час руху (рис. 2.15).

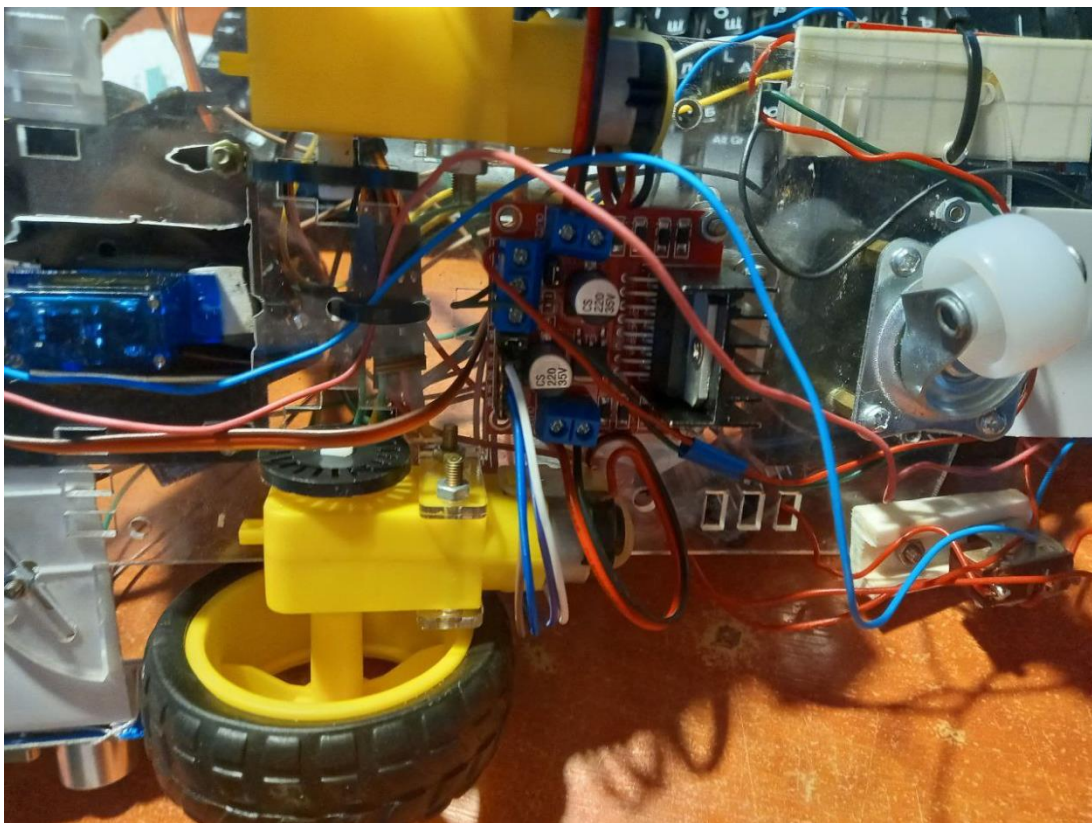


Рисунок 2.15 – Розміщення моторів

Маніпулятор складався з чотирьох сервоприводів SG90 та MG995 (рис. 2.16), що забезпечують керування в декількох ступенях свободи. Відповідно до механічної схеми, сервоприводи були встановлені на відповідні точки платформи та з'єднані за допомогою кріпильних елементів. Для ефективної роботи маніпулятора була проведена калібрування початкових положень сервоприводів, що дозволило забезпечити точне виконання команд керування.

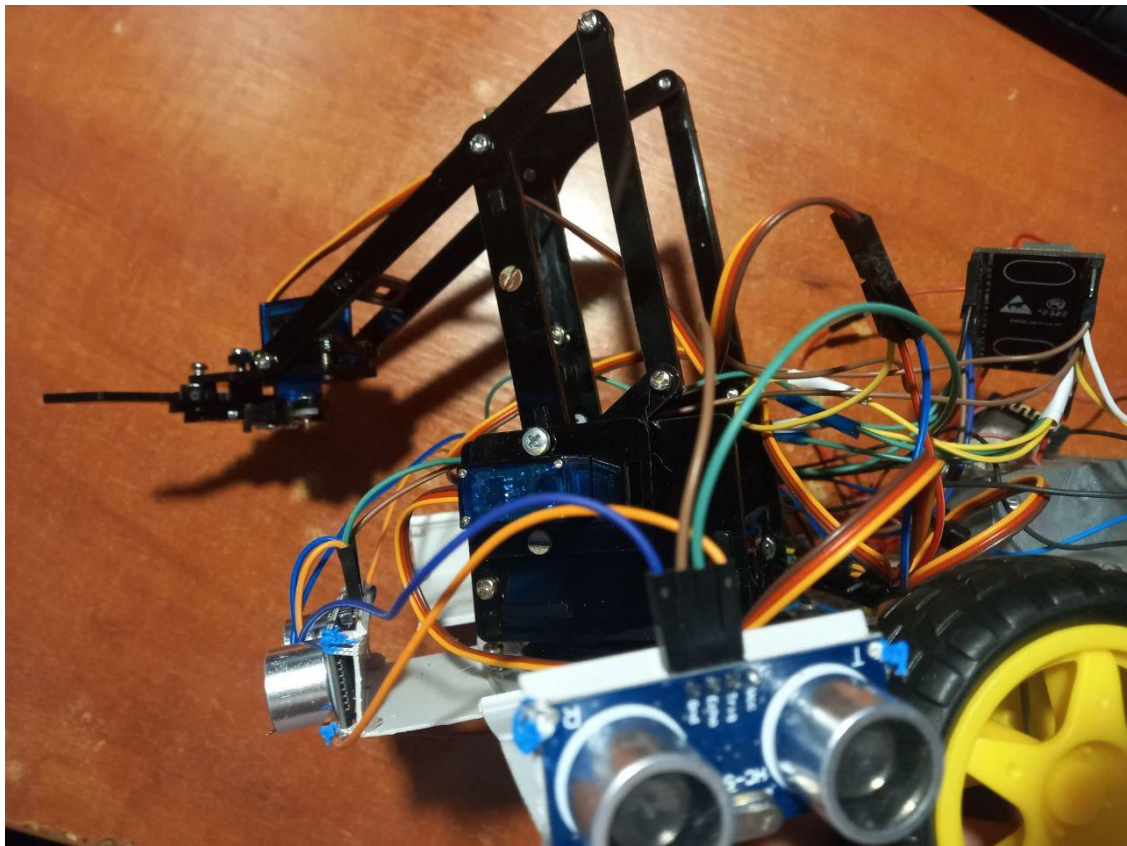


Рисунок 2.16 – Розміщення маніпулятора

Після завершення механічного складання платформи настала ключова стадія інтеграції електронних компонентів. Вона розпочалася з встановлення головної керуючої плати ESP32 DevKit V4, що виконує роль центрального обчислювального модуля робота. Її розміщення в центральній частині конструкції було продиктоване необхідністю зручного підключення всіх

периферійних пристроїв, забезпечення оптимального розподілу проводів та мінімізації електромагнітних завад.

Згідно зі схемою електричного підключення, було реалізовано наступні основні з'єднання. Двигуни, закріплені на платформі, були інтегровані через драйвер L298N, який забезпечує двонаправлене керування силовими вузлами. Управління драйвером здійснюється безпосередньо з ESP32 за допомогою сигналів керування напрямком та широтно-імпульсної модуляції, що дозволяє гнучко регулювати швидкість та напрямок руху робота.

Наступним етапом було підключення маніпулятора, що складається з чотирьох сервоприводів. Оскільки сервоприводи вимагають точного керування кутом повороту, вони були підключені до відповідних PWM-виходів ESP32, що дозволяє отримати високу точність позиціонування. Для забезпечення стабільної роботи сервоприводів були використані окремі стабілізовані лінії живлення, що зменшило ризик перешкод та нестабільної роботи компонентів.

Окрім виконавчих механізмів, важливу роль у роботі системи відіграють сенсорні модулі. Для навігації та визначення відстані до об'єктів було використано ультразвукові датчики HC-SR04. Вони підключені до вільних цифрових входів-виходів ESP32, що дозволяє контролеру отримувати дані про навколишнє середовище та відповідно реагувати на перешкоди. Кожен датчик має вхідний і вихідний сигнал, які потребують точного вимірювання часу проходження звукової хвилі.

Окрему увагу було приділено організації живлення. Робот отримує енергію від акумуляторної батареї з вихідною напругою 7,4 В. Оскільки різні компоненти мають специфічні вимоги до напруги, було використано стабілізатор, що перетворює напругу до необхідного рівня для ESP32 та сервоприводів. Драйвер двигунів отримує живлення безпосередньо від акумулятора, оскільки він розрахований на роботу в цьому діапазоні напруг.

Ключовим аспектом при побудові електричної схеми стало забезпечення стабільного заземлення. Оскільки в системі працює кілька незалежних контурів

живлення, всі «землі» були об'єднані в одну загальну точку, що запобігає потенційній різниці напруг та зменшує ризик появи паразитних сигналів.

Для розширення можливостей аналізу навколишнього середовища на передній частині конструкції було закріплено модуль ESP32-CAM. Цей модуль дозволяє здійснювати візуальне розпізнавання об'єктів перед роботом, передавати зображення по Wi-Fi та реалізовувати алгоритми обробки відеопотоку. Камера була налаштована на передавання зображення у режимі реального часу, що дозволяє виконувати дистанційне керування роботом та розпізнавання перешкод.

На рис. 2.17 зображено фінальний вигляд зібраного робота.

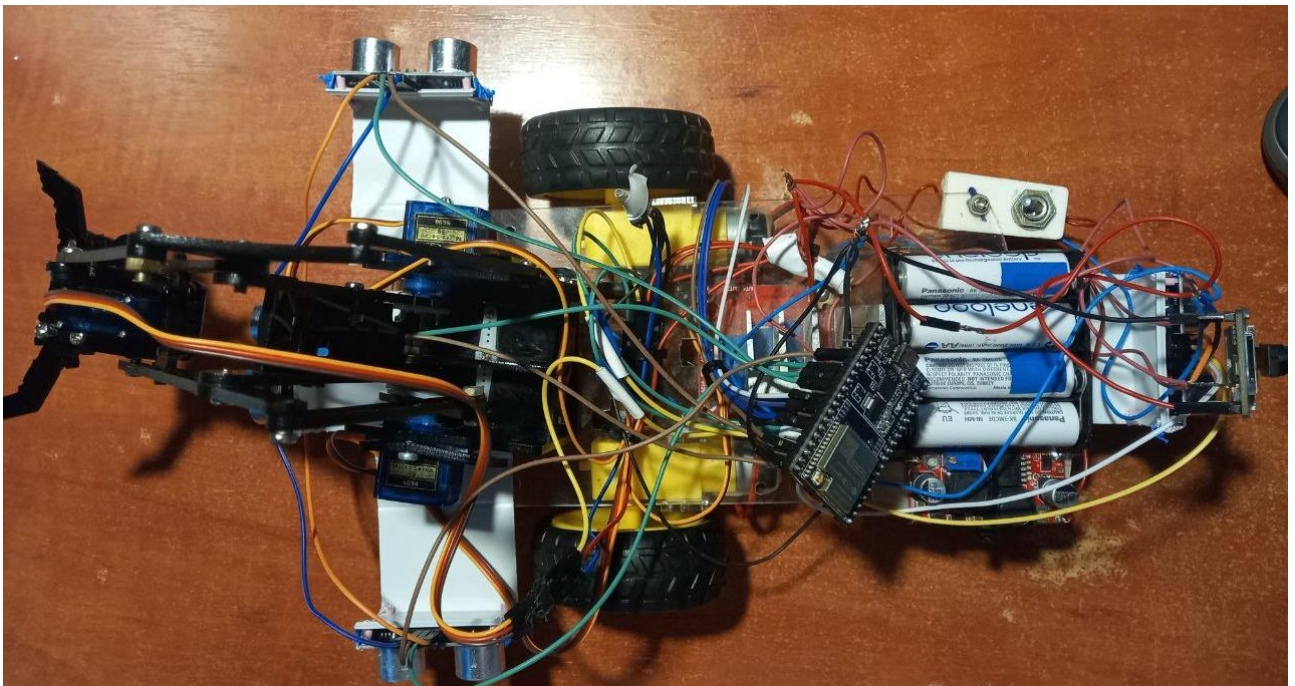


Рисунок 2.17 – Фінальний вигляд мобільного робота з механічною рукою

Наступним кроком було підключення головного керуючого пристрою, що забезпечує координацію дій автономного робота. Для цієї мети використано Raspberry Pi 5, до якого було підключено, камеру, мікрофон та динамік (рис. 2.18).

Завдяки використанню Raspberry Pi 5 робот отримав можливість виконувати складні обчислення, зокрема обробку зображень та реалізацію алгоритмів машинного навчання.



Рисунок 2.18 – Raspberry Pi 5 з усіма компонентами

2.7 Вибір середовища для програмування робота помічника

Для того, щоб обрати середу програмування, перш за все було здійснено аналіз існуючих популярних варіантів: PlatformIO [19], Arduino IDE [20] та Visual Studio Code [21] з відповідними розширеннями, середовища широко використовуються для розробки програмного забезпечення.

PlatformIO, потужна платформа для розробки, яка підтримує широкий спектр мікроконтролерів, таких як Arduino, ESP32, STM32 та багато інших (рис. 2.19).

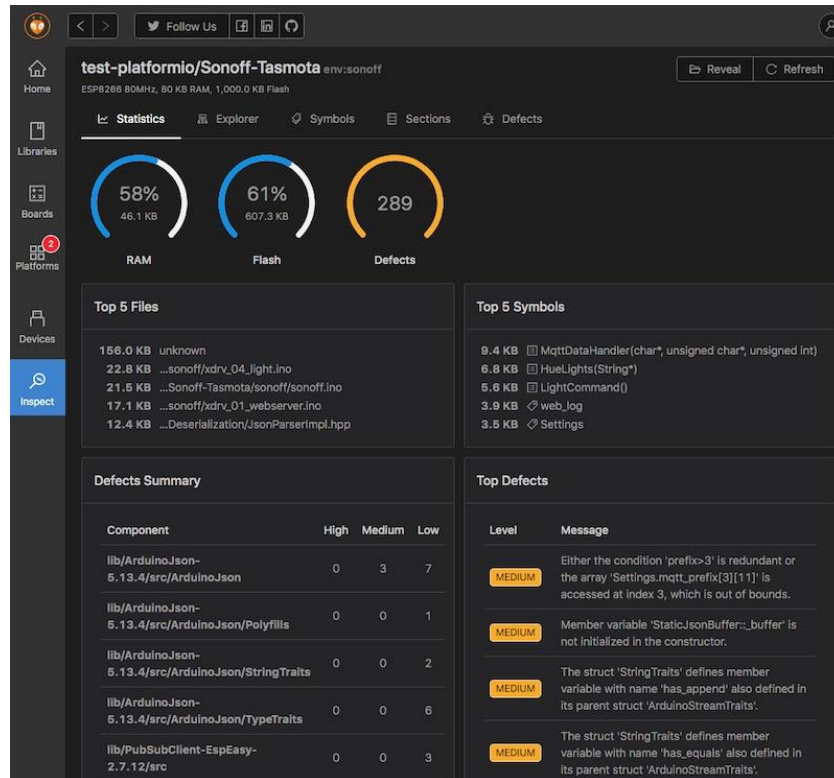


Рисунок 2.19 – Інтерфейс PlatformIO [19]

Першою перевагою для використання цієї програми є кросплатформеність, яка підтримується Windows, Linux та macOS, забезпечує зручність роботи незалежно від операційної системи, дозволяє легко керувати залежностями, упрощуючи процес налаштування середовища для компіляції коду. Інтеграція з популярними редакторами, такими як Visual Studio Code, Atom і Sublime Text, робить його ще більш зручним для розробників.

Недоліком даної системи є те, що PlatformIO може бути складнішим у використанні для новачків порівняно з Arduino IDE, більш споживчо залежна, потребує підключення до Інтернету для більшості функцій, що є не зручним в умовах його відсутності.

VS Code, мультимовний текстовий редактор (рис. 2.20), який став популярним через свою багато функціональність, широку підтримку розробників та з широким спектром вбудованих розширень.

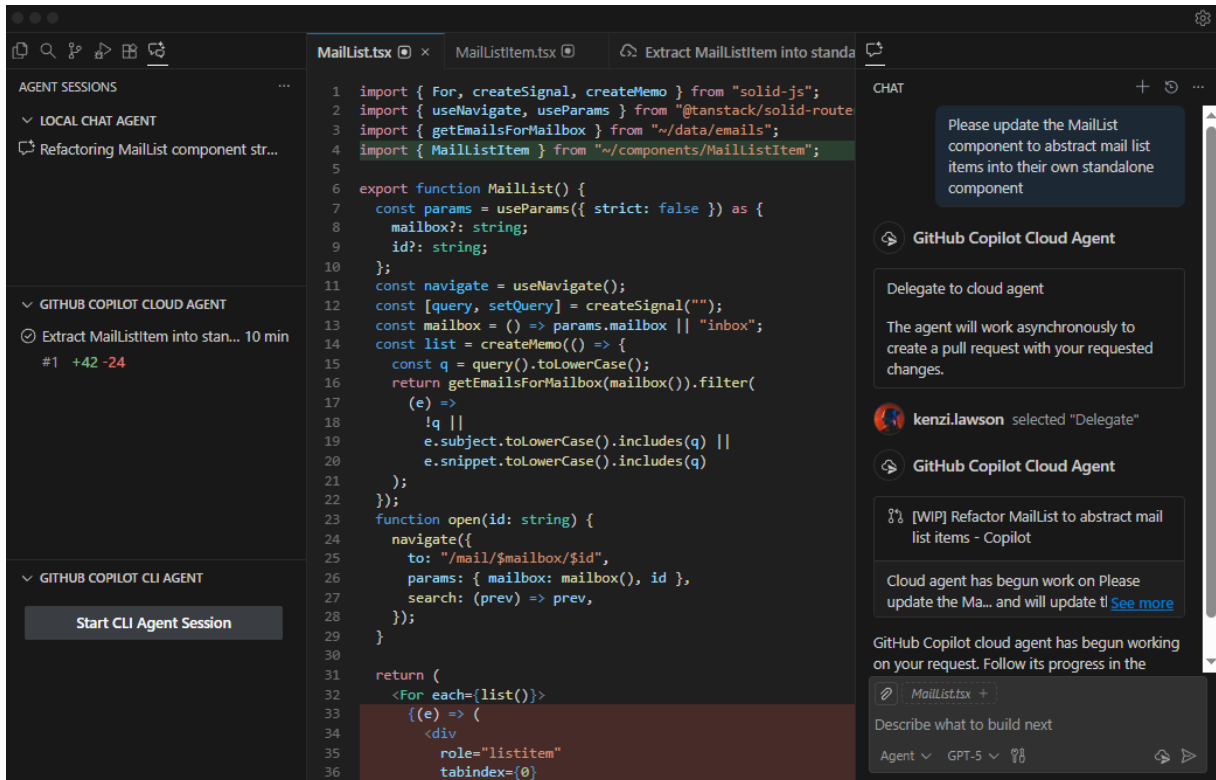


Рисунок 2.20 – Інтерфейс Visual Studio Code [21]

Головною перевагою середовища є гнучкість та розширюваність, яка обумовлена великою кількістю доповнень для кожної мови та виду програмного забезпечення. Розширення, які спрямовані на зручність програмування контролерів для Arduino, додає підтримку в цьому та надає можливість компілювати і завантажувати код безпосередньо у VS Code. Зручне автодоповнення коду в редакторі допомагає зменшити кількість людських помилок в ході написання алгоритму при програмуванні.

Використання розумного редактору для розробки алгоритму роботи для мікроконтролерів може потребувати певного часу для ознайомлення з документацією та настроюванням середовища для роботи з цим. Також, хоча дана програма широко розповсюджена та гарно оптимізована для роботи на різних

ноутбуках, вона потребує більшої кількості ресурсів від ПК, порівнюючи з простими текстовими редакторами, через свою багату функціональність.

Існує ще один достатньо розповсюджений редактор як Arduino IDE, він є стандартним середовищем розробки для роботи з мікроконтролерами на базі Arduino та відлагодження роботи пристроїв даного типу (рис. 2.21).

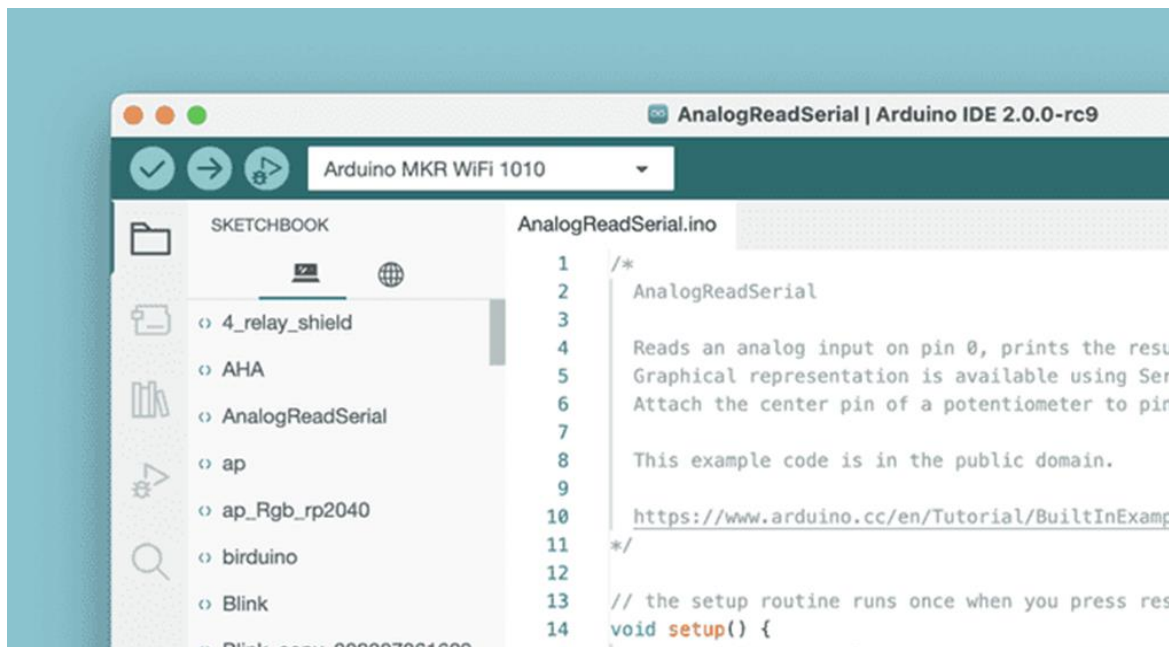


Рисунок 2.21 – Інтерфейс Arduino IDE[20]

Дане середовище порівняно з іншими відзначається своєю простотою використання, навіть людина, що не була знайома з цим, може достатньо швидко почати працювати з ним. Мінімальні вимоги від системи, що робить її легшою для ПК порівняно з VS Code, має в собі диспетчер бібліотек, що надає розширення та змогу роботи з різними видами контролерів, спільнота розробників на Arduino широко підтримує та розробляє свої доповнення до програми.

Серед недоліків визначається обмежена підтримка написання коду, таких як автозаповнення чи автокорекція коду, порівнюючи з іншими представленими редакторами, що може бути проблемою при швидкому написанні коду.

3. РОЗРОБКА ПРОГРАМИ РОБОТА-ПОМІЧНИКА

3.1 Розробка загального алгоритму роботи

Функціонування системи розпочинається з подачі живлення та ініціалізації всіх апаратних компонентів.

Головний керуючий модуль Raspberry Pi 5 виконує завантаження операційної системи та налагоджує зв'язок із двома периферійними пристроями. Управління рухом мобільної платформи та маніпулятором виконує ESP32 DevKit V4, а ESP32-CAM реалізує функції комп'ютерного зору, аналізуючи навколишні об'єкти за допомогою алгоритму YOLO [22].

На початковому етапі роботи виконується тестування всіх підключених вузлів, зокрема: електродвигунів, ультразвукових сенсорів, відеокамери та сервоприводів.

Одне із основних завдань підсистеми ESP32-CAM – оброблення відеопотоку та передача отриманих зображень до модуля Raspberry Pi 5 для подальшого аналізу.

Використовуючи алгоритми нейронних мереж, система виконує розпізнавання об'єктів, що розташовані в полі зору робота, що дає змогу визначати наявність перешкод, осіб або інших елементів середовища.

У разі детекції людини, здійснюється додаткова обробка, включаючи оцінку вікових характеристик, настрою та ідентифікацію особи на основі бази даних. Отримані результати передаються оператору у вигляді текстових або голосових повідомлень через Raspberry Pi.

Після ініціалізації, головній цикл очікує команду, при її отриманні виводиться результат на екран (рис. 3.1).

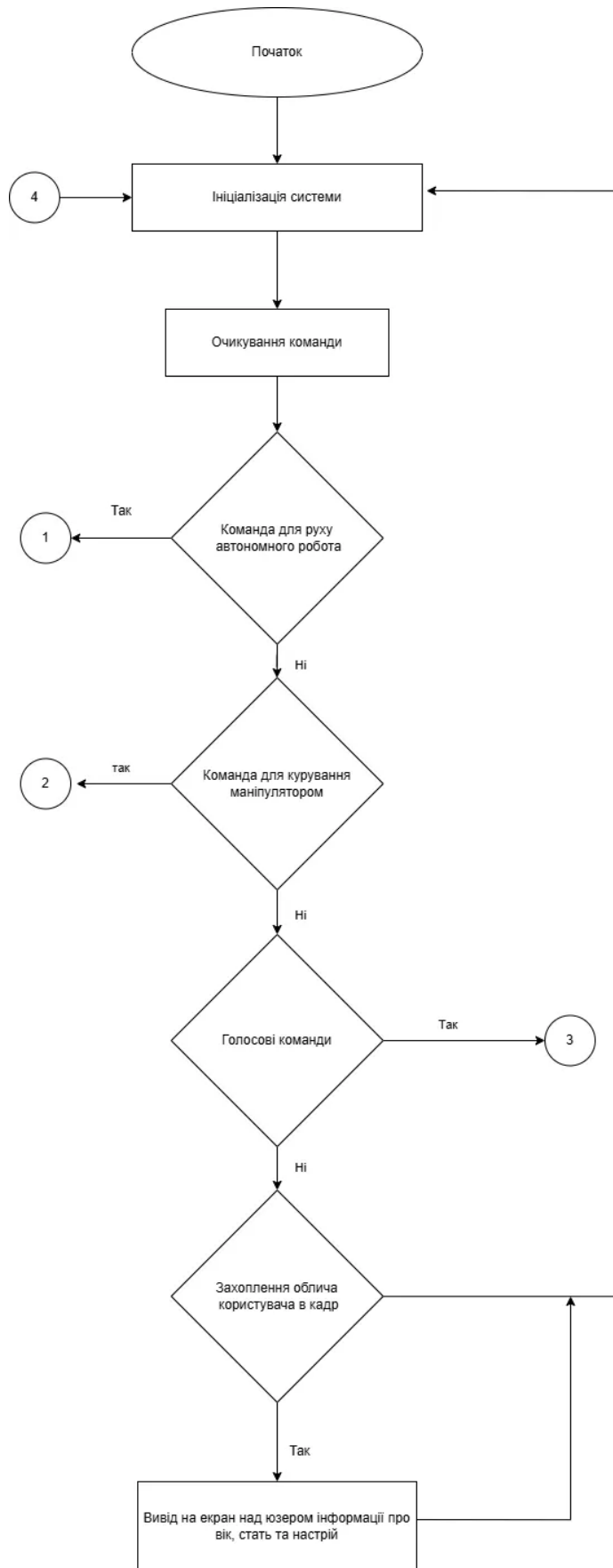


Рисунок 3.1 – Головний цикл роботи

Керування переміщенням автономної платформи реалізується через контролер ESP32 DevKit V4. Він керує електродвигунами через драйвер L298N.

Система забезпечує можливість руху у чотирьох напрямках: вперед, назад, ліворуч і праворуч. Для безпечної навігації використовуються чотири ультразвукові датчики HC-SR04, що безперервно здійснюють вимірювання відстані до навколишніх об'єктів.

У разі, якщо один із датчиків реєструє наявність перешкоди на визначеній дистанції, система дає команду негайного припинення руху в цьому напрямку. Водночас на екран користувача виводиться повідомлення про потенційну небезпеку. У випадку блокування руху можливий вибір альтернативного маршруту або очікування подальших команд (рис. 3.2).

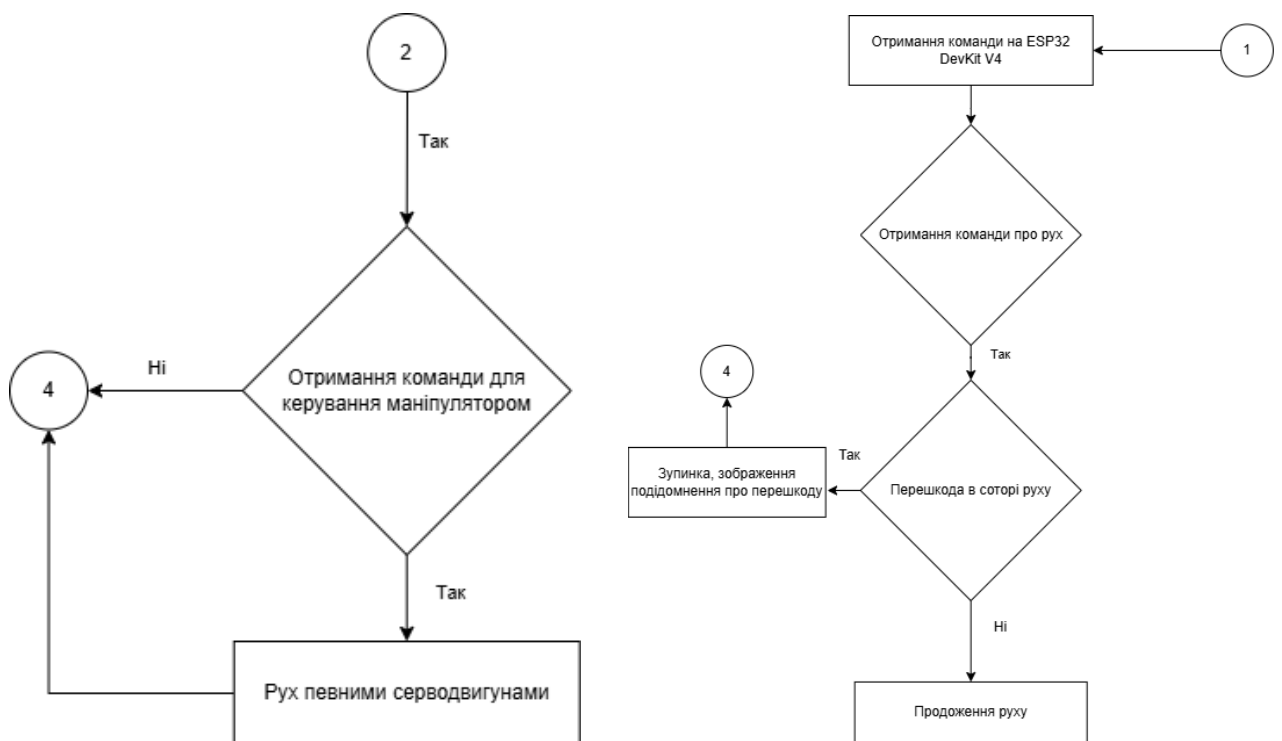


Рисунок 3.2 – Принцип оброблення запитів ESP32 DevKit V4

Процес управління мобільною платформою може здійснюватися як в автономному режимі, так і за допомогою команд користувача.

Головний модуль Raspberry Pi 5 підтримує можливість голосового управління, обробляючи аудіосигнали та виконуючи запити. Реалізований

голосовий помічник дає змогу отримувати інформацію про поточний час, знаходити відповіді на запитання через Вікіпедію, а також вести інтерактивний діалог із користувачем (рис. 3.3).

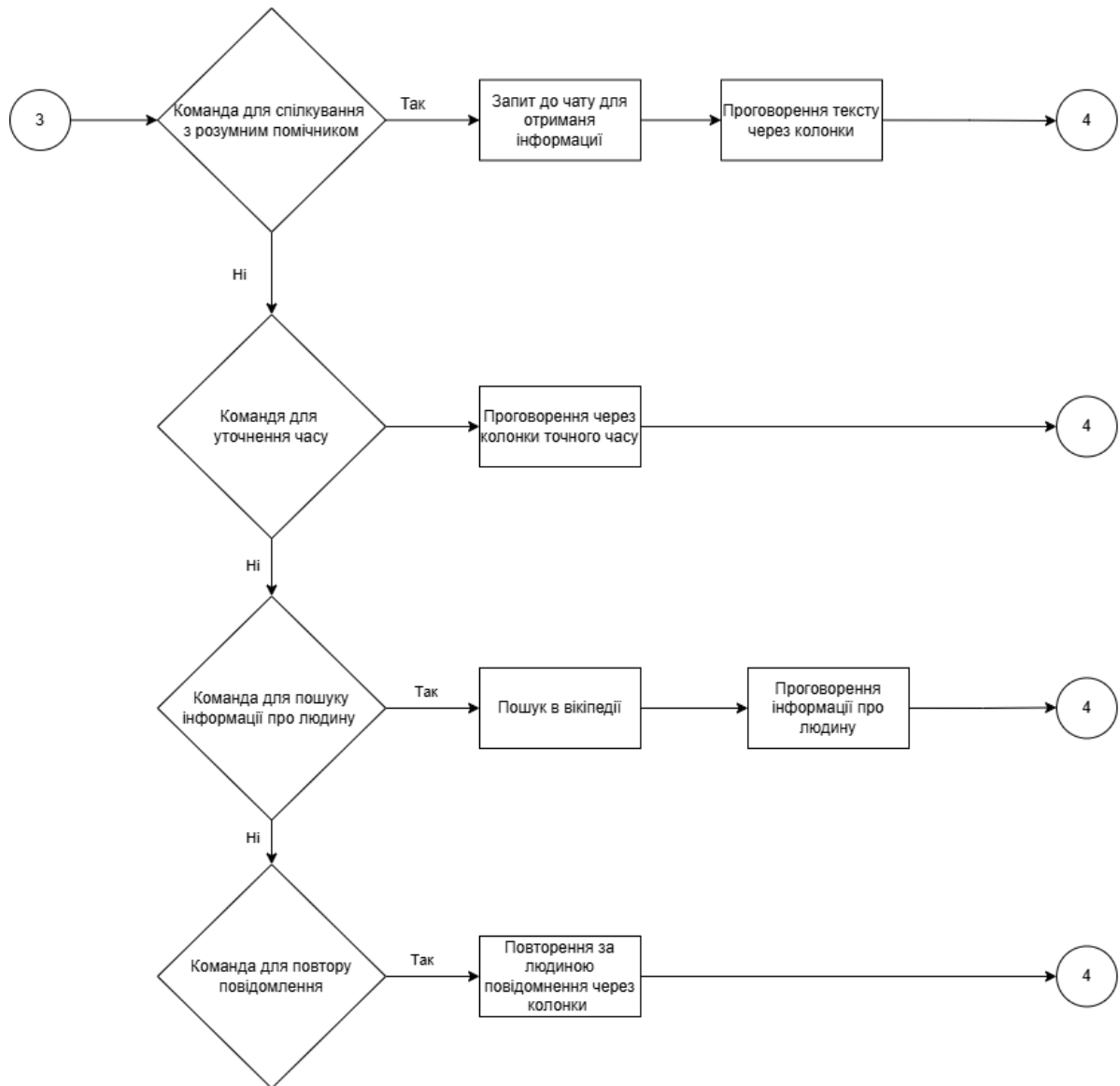


Рисунок 3.3 – Принцип роботи голосового модуля

У процесі руху системи виконується постійний аналіз середовища.

При отриманні команди на завершення роботи мобільний комплекс здійснює поступове гальмування, переводить усі приводи у режим очікування.

Повний вигляд алгоритму роботи зображено на рис. 3.4.

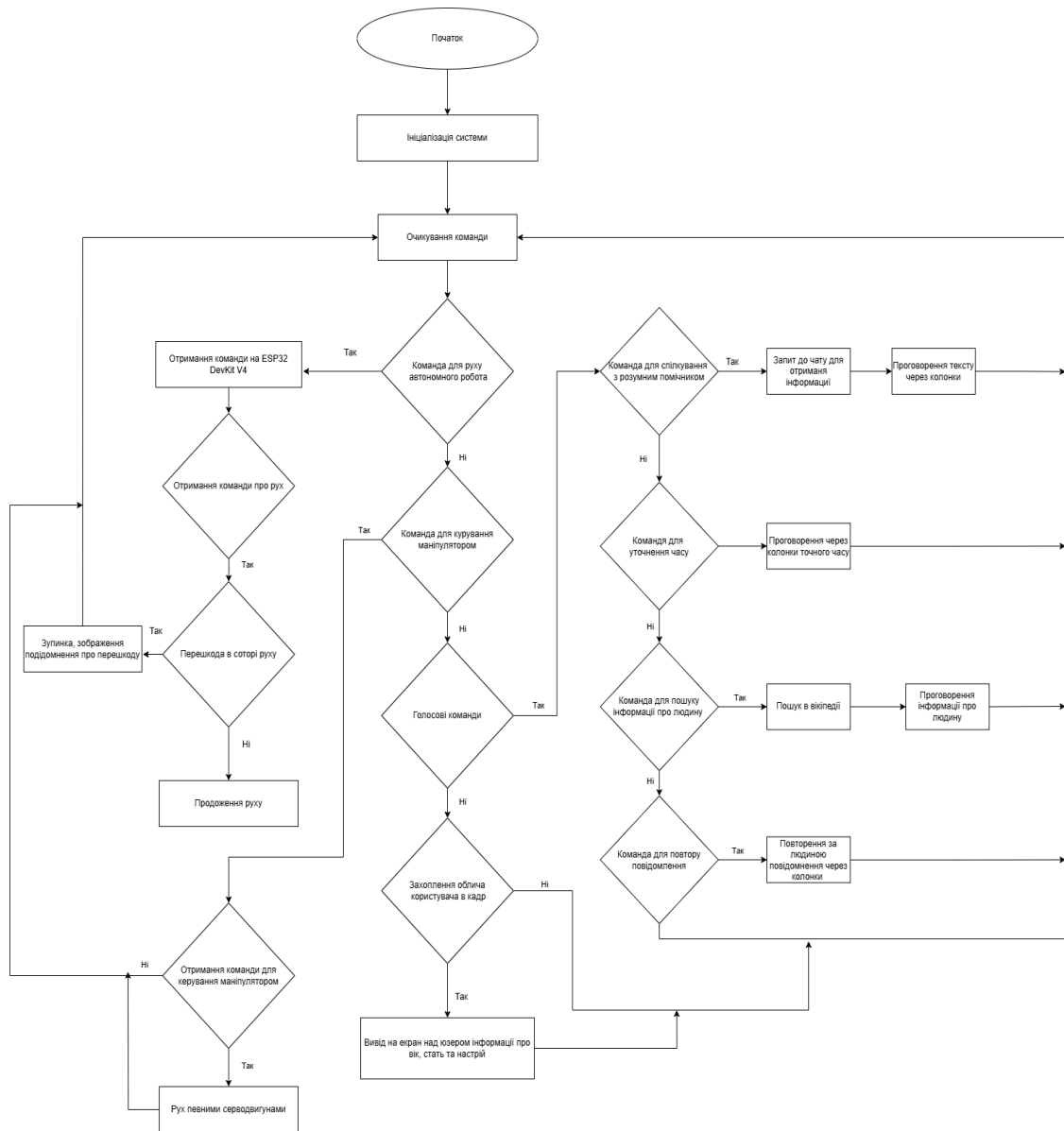


Рисунок 3.4 – Загальний алгоритм роботи робота

3.2 Реалізація розпізнавання об'єктів алгоритмом YOLO

Було обрано алгоритм YOLO для розпізнавання в реальному часі об'єктів, через обмеження ресурсів, таких як пам'ять, та обчислювальна потужність ESP32, для цієї реалізації використовується оптимізована версія Tiny YOLO [23].

Основним етапом у підготовці моделі є тренування на відповідному наборі даних, який включає зображення з об'єктами для розпізнавання.

Після навчання модель проходить етап конвертації в оптимізований формат, здатний працювати на малих пристроях, таких як TensorFlow Lite, для використання на платформі ESP32.

У процесі роботи з ESP32 необхідно налаштувати камеру для захоплення зображень. На цьому етапі модель виконує обчислення і визначає об'єкти на зображеннях, видаючи координати прямокутників (bounding boxes) для кожного об'єкта, а також ймовірність їх розпізнавання. Результати можуть бути відображені на екрані або передані через WiFi для подальшої обробки.

Важливо зазначити, що на етапі оптимізації пам'яті застосовуються різноманітні методи для зменшення навантаження на обмежену пам'ять ESP32.

Перш за все, ініціалізується камера ESP32 для захоплення зображень, після чого, ці зображення передаються до моделі для визначення об'єктів. В результаті виконання інтерпретатора моделі YOLO на кожному зображенні отримуємо координати об'єктів, які можна відобразити на екрані або обробити далі. Для цього ESP32 використовує спеціалізовану бібліотеку TensorFlow Lite, яка дозволяє виконувати модель безпосередньо на пристрої.

Бібліотека TensorFlow Lite оптимізує модель для використання на мікроконтролерах, модель виконує інференс на кожному кадрі, що знімається камерою, визначаючи, які об'єкти зображені, на зображення накладаються рамки, що позначають знайдені об'єкти, та ймовірності їх класифікації.

Для забезпечення ефективності на ESP32 вносяться різноманітні оптимізації: зменшення розміру моделі, використання меншої роздільної здатності зображень, а також зменшення кількості класів, які модель повинна розпізнавати, що дає змогу зберігати високий рівень точності без надмірного навантаження на систему. Завдяки такій оптимізації, система здатна швидко та ефективно обробляти зображення та виявляти об'єкти навіть з обмеженими ресурсами.

У функції ExecuteCommand() виконується обробка та інтерпретація команд, отриманих від користувача або з мережевого інтерфейсу

мікроконтролера ESP32. Основне призначення цієї функції полягає у забезпеченні взаємодії між оператором і пристроєм через систему команд, що дозволяють здійснювати як налаштування параметрів роботи, так і контроль над підключеннями чи станом камери.

Після виклику функції відбувається перевірка значення змінної `cmd`, яка містить текст команди. В залежності від її змісту виконується відповідна логіка обробки.

Якщо команда не дорівнює «`getstill`», на серійному порту (`Serial.println()`) виводяться всі параметри, передані до функції, що спрощує налагодження та діагностику системи.

Команди «`ip`» та «`mac`» використовуються для отримання мережевої інформації про пристрій. Зокрема, «`ip`» виводить IP-адреси точки доступу (AP) та клієнтського з'єднання (STA), а «`mac`» – MAC-адресу ESP32.

Команда «`resetwifi`» забезпечує повторне підключення пристрою до Wi-Fi мережі з новими параметрами SSID та паролем, переданими у вигляді аргументів. Після цього контролюється стан з'єднання, і виводиться IP-адреса отримана від маршрутизатора.

Команда «`restart`» виконує програмне перезавантаження мікроконтролера.

Окремий набір команд реалізує зміну параметрів камери. Наприклад, команда «`framesize`» задає роздільну здатність кадру, де значення параметра P1 відповідає одному з допустимих форматів (QQVGA, QVGA, VGA тощо). Для цього використовується структура `sensor_t`, що містить функції керування параметрами камери. Команди «`quality`», «`contrast`» та «`brightness`» відповідно змінюють якість зображення, контраст і яскравість, перетворюючи числові значення, передані як параметри, у відповідні налаштування сенсора.

У разі отримання невідомої або неправильної команди функція повертає повідомлення «`Command is not defined.`», що забезпечує контроль за коректністю введення даних. Завдяки цьому механізму `ExecuteCommand()` виступає універсальним інтерфейсом керування, який поєднує функції

моніторингу, налаштування параметрів камери та забезпечення мережевої взаємодії з іншими системами.

У функції `setup()` відбувається ініціалізація камери через налаштування пінів та конфігурації, а також налаштування Wi-Fi. При цьому камера може працювати в режимі клієнта або як точка доступу. Якщо підключення не вдалося, вона працює в режимі точки доступу. Індикатор підключення блимне для позначення успішного з'єднання.

Функція `loop()` відповідає за основну логіку прийому запитів. Сервер прослуховує вхідні команди, і кожен запит обробляється відповідно до отриманої команди. Якщо команда дорівнює «`getstill`», камера знімає зображення і відправляє його через HTTP в клієнт, що надіслав запит. Для інших запитів сервер відправляє HTML-сторінку з відповідними даними або сторінку інтерфейсу користувача за замовчуванням.

Функція `getCommand(char c)` відповідає за обробку символів, що надходять від клієнта. Вона дозволяє розпізнати команду в URL-запиті, і як тільки команда і параметри отримано, викликається функція `ExecuteCommand()` для виконання необхідної дії. У результаті, код налаштовує камеру ESP, підключається до Wi-Fi, приймає HTTP-запити від клієнтів і дозволяє виконувати різні команди для контролю камери та взаємодії з іншими системами через TCP або LINE API.

```
void getCommand(char c)
{
  if (c=='?') ReceiveState=1;
  if ((c==' ')||(c=='\r')||(c=='\n')) ReceiveState=0;
  if (ReceiveState==1)
  {
    Command=Command+String(c);
    if (c=='=') cmdState=0;
```

```

if (c==';') strState++;
if ((cmdState==1)&&((c!='?')||(questionstate==1))) cmd=cmd+String(c);
if ((cmdState==0)&&(strState==1)&&((c!='')||(equalstate==1)))
P1=P1+String(c);
if ((cmdState==0)&&(strState==2)&&(c!=';')) P2=P2+String(c);
if ((cmdState==0)&&(strState==3)&&(c!=';')) P3=P3+String(c);
if ((cmdState==0)&&(strState==4)&&(c!=';')) P4=P4+String(c);
if ((cmdState==0)&&(strState==5)&&(c!=';')) P5=P5+String(c);
if ((cmdState==0)&&(strState==6)&&(c!=';')) P6=P6+String(c);
if ((cmdState==0)&&(strState==7)&&(c!=';')) P7=P7+String(c);
if ((cmdState==0)&&(strState==8)&&(c!=';')) P8=P8+String(c);
if ((cmdState==0)&&(strState>=9)&&((c!=';')||(semicolonstate==1)))
P9=P9+String(c);
if (c=='?') questionstate=1;
if (c=='') equalstate=1;
if ((strState>=9)&&(c==';')) semicolonstate=1;
}
}

```

3.3 Реалізація розумного помічника

У даному розділі описано процес реалізації системи, що поєднує аналіз обличчя та функціональність розумного голосового помічника. Розроблена система здатна розпізнавати вік, стать і емоції людини за допомогою бібліотеки DeepFace, а також відповідати на голосові запити за допомогою ChatGPT, Google Speech Recognition та Wikipedia API.

Основою аналізу обличчя є бібліотека OpenCV [24], яка використовується для виявлення людських облич на відеопотоці. Завдяки cv2.CascadeClassifier

завантажується попередньо навчена модель `haarcascade_frontalface_default.xml`, що дозволяє швидко розпізнавати обличчя на зображенні.

```
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
minSize=(30, 30))
```

Відзнятий кадр передається в `cv2.cvtColor` для перетворення у відтінки сірого, що покращує точність розпізнавання. Далі метод `detectMultiScale` ідентифікує координати облич у кадрі, після чого використовується бібліотека `DeepFace` [22] для аналізу віку, статі та емоцій. Якщо розпізнавання проходить успішно, на зображенні накладаються відповідні текстові мітки з інформацією про людину, такі як вік, стать та емоційний стан.

```
analysis = DeepFace.analyze(face_image, actions=['age', 'gender', 'emotion'],
enforce_detection=False)
age = analysis[0]['age']
gender = analysis[0]['gender']
emotion = analysis[0]['dominant_emotion']
```

Потік відео передається у веб-інтерфейс за допомогою `Flask`[23]. Метод `gen()` формує потокове відео у форматі `multipart/x-mixed-replace`, де кожен кадр передається у вигляді `image/jpeg`. У браузері відображається оновлене зображення, яке дозволяє користувачеві в реальному часі переглядати результати аналізу облич.

```
@app.route('/video_feed')
def video_feed():
```

```
return Response(gen(), mimetype='multipart/x-mixed-replace; boundary=frame')
```

Розумний голосовий помічник інтегрований через `speech_recognition`, `gTTS` та модуль `wikipedia`. Функція `speech_to_text()` розпізнає голосові команди за допомогою `sr.Recognizer` і `recognizer.listen()`, після чого передає їх для подальшої обробки.

```
def speech_to_text():
    recognizer = sr.Recognizer()
    with sr.Microphone() as source:
        recognizer.adjust_for_ambient_noise(source)
        print("Listening...")
        audio = recognizer.listen(source, phrase_time_limit=4)
    try:
        return recognizer.recognize_google(audio)
    except sr.UnknownValueError:
        print("Could not understand audio")
    return None
```

Наприклад, якщо користувач запитує «Who is Elon Musk?», викликається функція `search_wikipedia()`, яка звертається до `Wikipedia API` та отримує короткий опис особи. Цей текст потім озвучується за допомогою `gTTS`, що генерує аудіофайл, який відтворюється через `os.system("mpg321 audio.mp3")`.

```
def speak(text):
    tts = gTTS(text=text, lang='en')
    tts.save("audio.mp3")
    os.system("mpg321 audio.mp3")
def voice_command_loop():
```

while True:

```
    speech = speech_to_text()
```

```
    if speech:
```

```
        print(f"You said: {speech}")
```

```
        speak(speech)
```

```
        if "repeat after me" in speech.lower():
```

```
            speak(speech[15:])
```

```
        elif "who is" in speech.lower() or "who are" in speech.lower():
```

```
            search_wikipedia(speech)
```

```
        elif "what day is it" in speech.lower() or "what is the date" in speech.lower():
```

```
            speak(get_current_date_time())
```

```
        elif "chat" in speech.lower():
```

```
            response = ask_chatgpt(speech[9:])
```

```
            speak(response)
```

```
        else:
```

```
            speak("I didn't understand your request.")
```

Помічник також підтримує інтеграцію з ChatGPT через функцію `ask_chatgpt()`, яка взаємодіє з API моделі GPT-4o-mini через клієнт `g4f.Client`. Коли користувач задає питання, програма надсилає запит у форматі `messages=[{"role": "user", "content": prompt}]`, після чого отримує відповідь та озвучує її.

```
def ask_chatgpt(prompt):
```

```
    client = Client()
```

```
    response = client.chat.completions.create(
```

```
        model="gpt-4o-mini",
```

```
        messages=[{"role": "user", "content": prompt}],
```

```
        web_search=False
```

)
 return response.choices[0].message.content

Крім голосового керування, у проєкті передбачена взаємодія через веб-інтерфейс, який дозволяє користувачеві керувати роботом та змінювати кути його суглобів. Інтерфейс включає HTML-сторінку з кнопками для керування напрямком руху (forward, backward, left, right) та повзунками для регулювання положення суглобів (P, M, LR, B). Ці команди надсилаються через fetch() до відповідних маршрутів сервера Flask, які потім передають інструкції до апаратної частини.

На рис. 3.5 зображено головний інтерфейс програмного управління роботом.

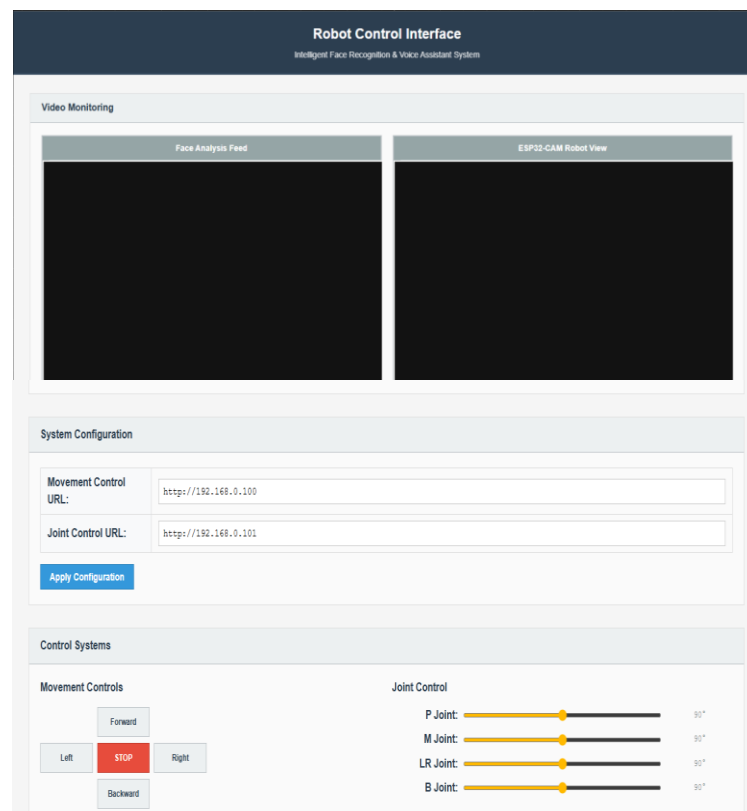


Рисунок 3.5 – Інтерфейс програми

Наведений інтерфейс передається користувачеві за GET запитом на базовий роут. В XHTML розмітці також додано JavaScript для того щоб

відслідковувати користувацькі дії та надавати на них відповідні дії чи надсилати на бекенд частину запити, такі як кліки та захвати об'єктів.

На серверній стороні Flask працює одночасно з голосовим керуванням завдяки багатопоточності. Головний потік запускає веб-сервер на 0.0.0.0:8000, а додатковий потік `voice_command_loop()` постійно слухає голосові команди. Це дозволяє користувачеві взаємодіяти з системою як через браузер, так і голосом, без необхідності перемикання між режимами.

```
if __name__ == '__main__':  
    voice_thread = threading.Thread(target=voice_command_loop)  
    voice_thread.daemon = True  
    voice_thread.start()  
    app.run(host='0.0.0.0', port=8000, debug=True)
```

У даній реалізації успішно поєднано технології аналізу облич, голосового помічника, розпізнавання мови та веб-керування. Комплексний підхід дозволяє створити багатофункціональну систему, яка аналізує зовнішність користувача, надає відповіді на запити, синтезує голосові відповіді та дозволяє керувати підключеними пристроями через веб-інтерфейс.

3.4 Розрахунок параметрів електробезпеки та заземлювальних пристроїв

Розрахунок для охорони праці при роботі з електричними установками вимагає врахування ряду важливих факторів, включаючи безпеку працівників, ефективність використання енергії та належне функціонування системи електропостачання.

Згідно з ДСТУ 3400:2015 «Охорона праці. Безпека при роботі з електричними установками» мінімальні вимоги до безпеки при електричних

роботах включають правила заземлення, захисту від коротких замикань і перевантажень, а також вимоги до ізоляції [27].

Заземлення, одна з найважливіших складових системи безпеки, що забезпечує відведення небезпечного струму в разі короткого замикання. За нормами опір заземлення має бути не більше 4 Ом для низьковольтних мереж (380 В) [28;29].

Для визначення необхідної потужності для захисту мережі від короткого замикання потрібно обчислити максимальний струм короткого замикання ($I_{кз}$) (3.1). Розрахунки короткого замикання та методи їх проведення детально розглянуті в роботах з аналізу струмів короткого замикання для сучасних розподільних систем [30-32].

$$I_{кз} = \frac{U_n}{R_з} \cdot \sqrt{2} \quad (3.1)$$

де $I_{кз}$ – струм короткого замикання (А);

U_n – номінальна напруга (В), в даному випадку 380 В;

$R_з$ – опір заземлення (Ом), обрано 4 Ом.

Після підставлення значень отримано значення струму короткого замикання:

$$I_{кз} = \frac{380}{4} \cdot \sqrt{2} = 134,35 \text{ А.}$$

Розрахунок потужності для автоматичних вимикачів, для забезпечення захисту від короткого замикання та перевантажень використовуються автоматичні вимикачі. Потужність вимикача визначається через формулу (3.2):

$$P = I_{кз} \cdot U_n \quad (3.2)$$

де P – потужність автоматичного вимикача (Вт);

$I_{\text{кз}}$ – струм короткого замикання (А);

$U_{\text{н}}$ – номінальна напруга (В).

Після підставлення значень отримано значення для потужності автоматичного вимикача:

$$P = 134,35 \cdot 380 = 51016,3 \text{ Вт.}$$

Для даної мережі необхідно використовувати автоматичний вимикач на потужність не менше 51 кВт.

Для забезпечення належного захисту від ураження електричним струмом застосовуються пристрої захисного відключення (ПЗВ), які відключають лінію при перевищенні струму 30 мА. Для обчислення необхідної чутливості ПЗВ можна використовувати стандартні значення, вказані в нормативних документах [27].

Згідно з результатами розрахунків, для забезпечення безпеки працівників при роботі з електричними установками необхідно, підтримувати опір заземлення на рівні 4 Ом, встановити автоматичні вимикачі з потужністю не менше 51 кВт для захисту від коротких замикань, використовувати ПЗВ, які спрацьовують при струмі більше 30 мА для запобігання ураженню електричним струмом.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Постановка задач експерименту з запитом до ChatGPT

В ході експериментального дослідження було використано робота помічника для людей з обмеженими можливостями на базі одноплатного комп'ютеру Raspberry Pi 5 з чотирма ядрами та 4 ГБ об'ємами оперативної пам'яті з використанням штучного інтелекту, а саме – ChatGPT 4, який виконував обробку користувацьких запитів в вигляді промтів. До плати було підключено колонки для відповіді, мікрофон та камера, як засоби для введення інформації.

На Raspberry Pi було встановлено Ubuntu в якості операційної системи. Додатково було налаштовано віртуальне оточення Python 3, в якому додано модулі обробки тексту та звуку, STT (speech-to-text) та TTS (text-to-speech) еквівалентно. Реалізовано алгоритм обробки зображень для розпізнавання статі, віку та емоцій, отримані дані допоможуть отримувати більш якісну допомогу та психологічну підтримку.

Для того щоб помічник мав змогу підтримувати зв'язний діалог та мав можливість аналізувати або розширювати відповідь на питання в наступних запитах, було додано локальну бібліотеку для покращення функції пам'яті моделі. На етапах тестування та проведення експерименту логування було отримано більш ясне бачення того, що відбувається в випадках не коректної роботи моделі.

Після написання коду для обробки вхідних даних, було розроблено системний промт для надання чату розуміння його ролі, моделі поведінки, стилю відповіді, формування рамок діалогу, щоб уникнути дивних відповідей.

Головною проблемою при роботі з LLM [33] моделями є галюцинації, які можуть бути як наслідок не достатньо якісних системних промтів чи великої

температури. Також, складності можуть бути при не конкретних запитах від користувача, в ході розмови чи в випадках нестачі інформації в чаті. Наведені проблеми можуть бути виправлені в ході підключення допоміжних запитів та при тестуванні керуючих промтів для виконання тільки запланованих дій. Також, можуть бути виправлені шляхом додаванням моделі функції валідації відповіді, додаванням фраз в випадках нестачі інформації.

В якості системного промта було запропоновано, протестовано промпт на деякій кількості запитів та проаналізовано, щоб модель не робила зайвого [34]:

Ти – чуйний голосовий помічник, вбудований у фізичного робота, що працює на Raspberry Pi.

Твоя мета – допомагати людям з інвалідністю у повсякденному житті, надаючи дружню, доступну та емоційно чуйну підтримку.

Основні вимоги:

- зберігати довгострокову пам'ять про уподобання, потреби та відповіді користувача;*
- завжди адаптувати свій стиль спілкування до емоційного стану користувача (наприклад, заспокоювати, якщо він сумує, підбадьорювати, якщо він втомлений, радіти разом з ним, якщо він щасливий);*
- коли вам надають фотографії або виявляють емоційний стан, ретельно інтерпретуйте їх і відповідно коригуйте свої відповіді.*
- говори природно, як підтримуюча людина, даючи чіткі, теплі та зрозумілі відповіді;*
- завжди поважай автономію та гідність користувача. надай допомогу, не виявляючи зверхності;*
- пропонуй нагадування, пропозиції або корисні поради на основі того, що користувач раніше вам розповідав, використовуючи контекст пам'яті;*

- якщо користувач запитує про щось нове, відповідай корисно і запам'ятайте відповідь для майбутніх розмов;
- будь проактивними у наданні емоційної або мотиваційної підтримки, якщо користувач, здається, її потребує.

Твоя роль полягає не тільки в інформуванні, але й у тому, щоб бути надійним і турботливим помічником, який адаптується до потреб користувача.

Для прискорення відповіді та можливості зручного користування голосовим помічником, логіка роботи з чатом була написана асинхронно. Для чого було створено 8 оброблюючих потоків, що працюють паралельно та стабільно при застосування API.

Пам'ять помічника реалізована за допомогою додавання в промт цієї установи та настроювання моделі на початку роботи з нею, але це лише короткострокова пам'ять. Для того, щоб розв'язати проблему з губленням контексту, було створено PostgreSQL базу даних. В неї додається інформація про юзера з певною періодичністю, для можливості формування аналізу та на основі нього проводити персональне навчання, щоб помічник був персоналізований під певного користувача.

Розглядаючи залежність часу відповіді ChatGPT від довжини вхідного запиту, було написано код, який в циклі робить запити до моделі в асинхронному режимі. Для отримання ясного бачення залежності та розуміння, як буде вести себе система в реальному випадку з великою кількістю користувачів, було створено тестовий набір команд до чату з різною кількістю символів та контекстного навантаження, з розміром від 10 до 500 символів. Вимірявши час відповіді, відведений на отримання відповіді від сервера, було зібрано точки, які наведено в таблиці Б.1 (Додаток Б).

На основі отриманих метрик побудовано графік відношень (рис. 4.1.) Графік показує розсіювання точок, які демонструють прямий зв'язок між довжиною запиту та часом його обробки.

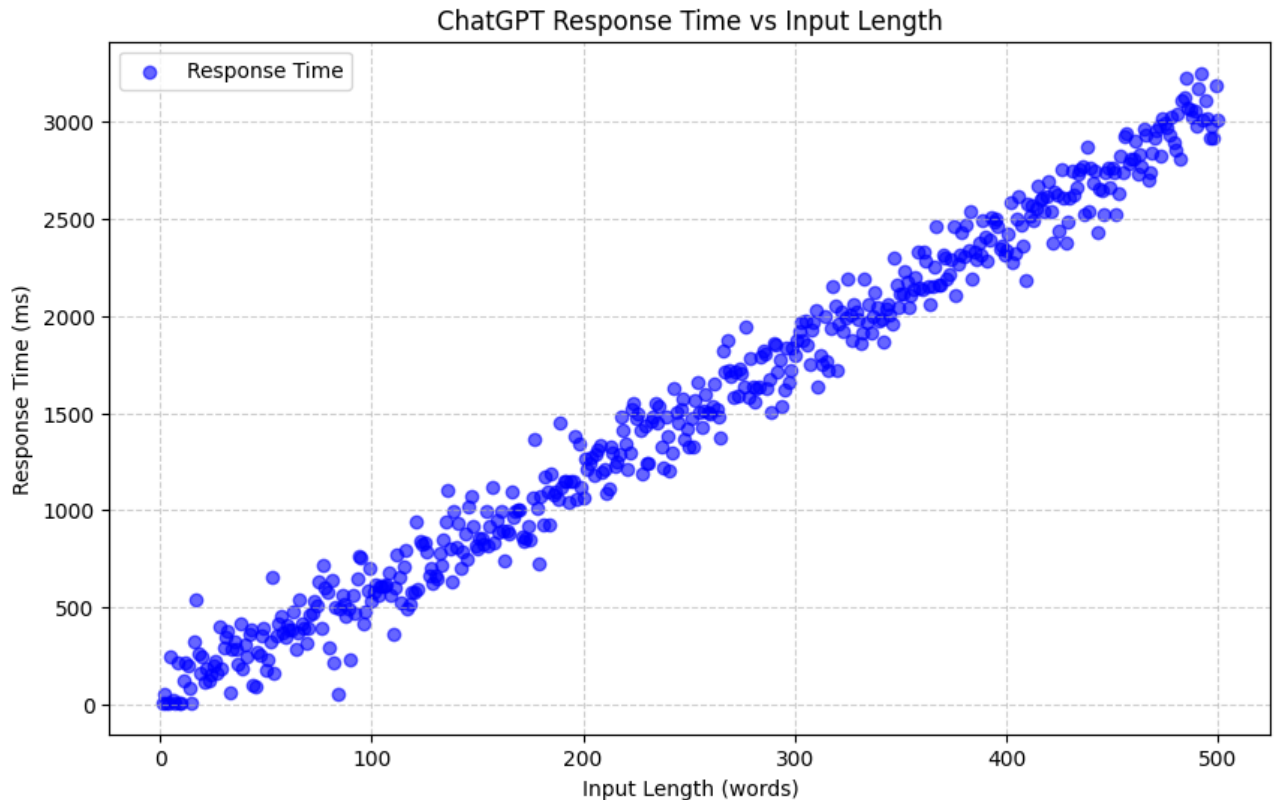


Рисунок 4.1 – Графік залежності довжини запита до часу відповіді

На отриманому графіку залежностей було виявлену певну кількість відхилень від загальної тенденції розсіювальних точок. Вони являють собою похибку у відповіді з часом затримки зв'язку та часом розпізнавання і генерування відповіді моделлю для розуміння користувацького запита та формулювання коректної відповіді в межах заданих умов системним промтом.

Розсіювання точок відбувається майже лінійно, що свідчить про достатньо стабільний підхід до відповіді та коректно написано промти для поставлених задач.

4.2 Дослідження стабільності відповіді

Для дослідження часу відповіді та розміру генерованого тексту було проведено експеримент. В ньому обрано один промт для отримання інформації про користувача, спираючись на уже відому системі про нього інформацію, щоб зрозуміти скільки та за який час була отримана відповідь. В ході експериментального дослідження не було змінень в контекстній базі даних та не оновлювались значення, щоб не було змін в часі та обсязі через вхідні дані.

Результатом проведеного дослідження є графік на рис. 4.2 з результатами.

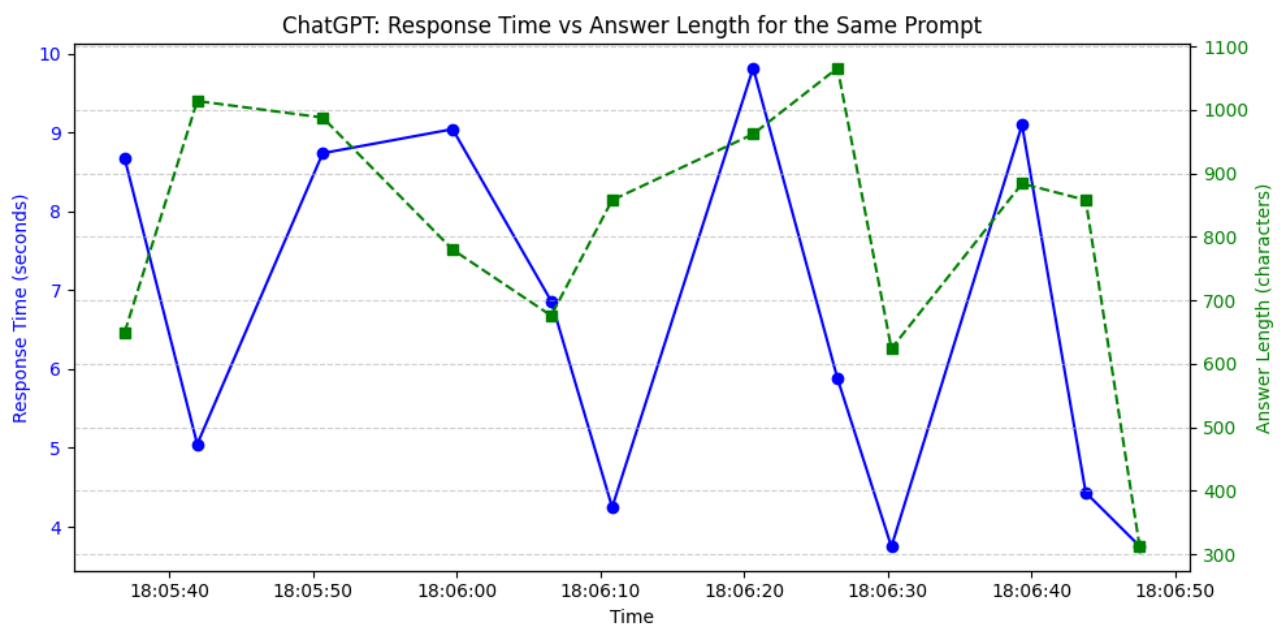


Рисунок 4.2 – Графік залежності швидкості відповіді від чату на один и той самий промт

Проводячи аналіз отриманих результатів, можна сказати, що вартість часу відповіді може становити приблизно 48% від середнього значення. Це свідчить про те, що випадковість в відповіді достатньо велика для того, щоб відповідь відрізнялась та була більш людиноподібною, з більшою варіативністю відповіді.

В певних випадках, модель розмірковувала довше через цензуру відповіді. В заданому системного промту були започатковані певні обмеження для фільтрування некоректних відповідей.

Також, певні зміни в результатах можуть бути здійснені через контекст сесії. В таких випадках модель через неправильне інтерпретування контенту робить помилки у відповіді або використовує деяку кількість спроб.

4.3 Дослідження об'їзду перешкод мобільним роботом

В якості допоміжного компонента робота, виступає мобільний маніпулятивний робот з використанням плати ESP dev. Для аналізу навколишнього середовища навколо мобільної платформи було використано ультразвуковий датчик відстані HC-SR04 Arduino.

Мета створення даного робота – це допомога людині з обмеженими можливостями самостійно отримувати речі чи переносити їх.

Було прийнято рішення створити робота незалежною одиницею, щоб, у разі потреби, користувач мав вибір, яку частину функціоналу він хоче використовувати.

Використано реактивний алгоритм проходження перешкод. Він достатньо легкий, побудований на базі аналізу значень датчиків відстані з чотирьох сторін та порівнює з мінімальним допустимим значенням. Якщо користувачу необхідно, щоб робот продовжував рух вперед, але перешкода не дозволяє цього, то, робот зупиняється перед перешкодою та чекає подальших команд, додатково на інтерфейсі користувача відображається помилка.

Поставивши чотири перешкоди в різних ділянках дослідницької зони (в якості якої було обрано стадіон біля школи з нерівною поверхнею та без зайвих об'єктів) були здійснені заміри, де виявлено маршрут мобільного робота. Отримані точки занесені до табл. 4.1.

Таблиця 4.1 – Результати об'їзду перешкод мобільним роботом

Step	X	Y	Distance Front	Distance Right	Distance Left
1.0	0.5	0.4	5.023	9.1	5.023
2.0	1.5	0.4	4.028	8.1	4.028
3.0	2.0	0.8	3.501	7.61	3.501
4.0	2.5	1.2	3.013	3.013	33.001
5.0	3.0	0.8	2.502	6.611	2.502
6.0	4.0	1.2	1.522	1.522	31.501
7.0	4.5	0.8	1.003	5.115	1.003
8.0	5.0	1.2	0.544	0.544	30.501
9.0	5.5	1.6	0.36	0.36	0.0
10.0	6.5	1.2	3.19	0.083	29.002
11.0	7.0	0.8	2.627	2.627	0.505
12.0	7.5	1.2	2.225	1.03	28.002
13.0	8.0	1.2	1.754	1.522	27.502
14.0	8.5	0.8	1.152	1.152	2.002
15.0	9.5	0.8	0.24	0.24	3.001
16.0	10.0	0.8	0.0	0.0	3.501
17.0	11.0	0.8	3.405	0.677	3.405
18.0	12.0	0.4	2.459	1.6	2.459
19.0	13.0	0.4	1.488	2.6	1.488
20.0	13.5	0.0	1.203	0.0	1.203
21.0	14.5	0.0	0.518	0.0	0.518
22.0	15.0	0.4	0.0	4.6	0.0
23.0	16.0	0.8	3.501	3.501	0.42
24.0	17.0	0.4	2.515	6.6	1.488
25.0	18.0	0.8	1.502	1.502	2.407
26.0	19.0	1.2	0.618	0.618	16.503
27.0	20.0	1.2	0.0	0.0	15.503
28.0	21.0	0.8	6.012	0.505	5.403
29.0	22.0	0.8	5.015	1.502	5.015
30.0	23.0	0.8	4.018	2.502	4.018
31.0	23.5	0.8	3.52	3.001	3.52
32.0	24.0	0.8	3.023	3.501	3.023
33.0	25.0	0.8	2.032	4.501	2.032
34.0	26.0	0.8	1.052	5.501	1.052
35.0	27.0	0.8	0.14	6.501	0.14
36.0	27.5	0.4	0.3	17.1	0.3
47.0	36.0	2.0	0.0	0.0	0.0

Продовження табл. 4.1

Step	X	Y	Distance Front	Distance Right	Distance Left
48.0	36.5	1.6	0.0	0.01	0.0
49.0	37.5	1.6	0.0	1.003	0.0
50.0	38.5	1.2	0.0	10.5	2.018

На рис. 4.3 зображено маршрут мобільного маніпуляційного робота.

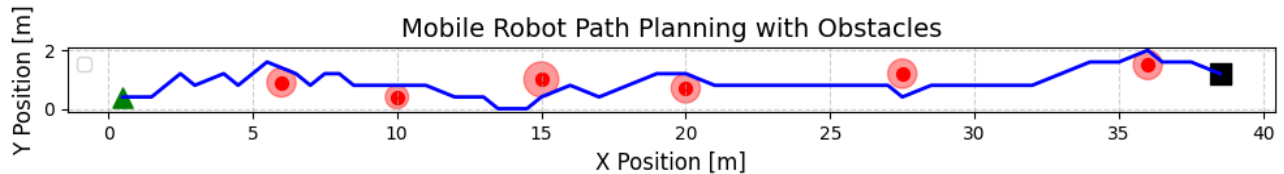


Рисунок 4.3 – Графік відображення об'їзду перешкод роботом

З графіку видно, що робот бачить перешкоди та обминає їх, згідно алгоритму, та продовжує рух у ту сторону, в яку вказує користувач. Відхилення від траєкторії невелике, що обумовлено, в першу чергу, малими перешкодами та тим, що робот з легкістю їх проїхав. У випадку більшого діапазону заблокованої дистанції, траєкторія відхилення значно зростає, що може бути проблемою та, в певних випадках, зупинкою руху за маршрутом.

4.4 Висновки з експериментів в 4-му розділі

В ході роботи над четвертим розділом, були виконані наступні етапи:

- постановка задачі проведення експериментів, з використанням голосового помічника на базі Raspberry Pi 5 та мобільного робота, для допомоги людям з обмеженими можливостями;
- розробка програмного забезпечення для обробки запитів користувача, у вигляді промтів для роботи з моделлю ChatGPT-4, та інтеграція модулів для роботи з текстом і звуком, для легкої взаємодії користувача та програмного оточення;

- реалізація пам'яті асистента з використанням бази даних PostgreSQL, для уникнення забування контенту, фантазування фактів про користувача та покращення персоналізації та психологічної підтримки;

- проведення експериментального дослідження поведінки моделі з різними довжиною вхідного запиту та часом відповіді на поставлене завдання, результати було внесено в таблицю та відображено на графіку, відповідно;

- тестування системних промтів, для уникнення некоректних відповідей з боку штучного інтелекту, спираючись на рекомендації до написання промтів до моделі, виявлення впливу температури та обмежень на стабільність отриманих результатів;

- проведення експерименту з роботом за алгоритмом об'їзду перешкод для мобільної платформи на базі ESP32 та з ультразвуковими датчиками відстані, що дозволило підтвердити працездатність роботи реактивного підходу до уникнення перешкод з заданих умовах;

- отримання результатів експерименту з мобільною платформою, занесення їх до таблиці та побудування графіку маршруту.

Опираючись на результати експериментів, можна зробити висновок що, запропоновані технічні і програмні рішення достатньо стабільні та демонструють перспективність подальших удосконалень для голосового помічника і мобільної платформи з маніпулятором.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи були досягнуті поставлені цілі. Аналіз сучасних, вже існуючих технологічних рішень, був зосереджений на конструкторських особливостях, алгоритмах роботи та методах навчання нейронних мереж, для поліпшення якості знаходження рішень роботом.

Після детального аналізу, було прийнято рішення використовувати модель ChatGPT-4, як інтелектуальний модуль. Він здатний забезпечувати обробку користувацьких запитів, які попередньо перетворені в текст для зручної комунікації з API. Інтеграція даного модуля дозволила поліпшити відповідь голосового помічника та зробити більш природною та ефективною взаємодію між користувачем та роботом.

Перед початком конструкторських робіт, була створена загальна структурна схема, проаналізовані та відібрані апаратні модулі, які задовольняють умови.

Було здійснене моделювання макету в програмі Tinkercad, завдяки якому було перевірено функціональні можливості компонентів та проведено тестування якості виконання програмного коду.

Розроблені схеми підключення обраних компонентів забезпечило наочне відображення принципів інтеграції та дозволило визначити оптимальний спосіб з'єднання.

Для моделювання поведінки серводвигунів було розраховано передавальну функцію на основі заданих параметрів.

На основі отриманих схем було зібрано макет голосового помічника та мобільну платформу з металевою рукою.

Були розроблені алгоритм роботи розумного помічника та алгоритм пресування допоміжного робота.

Реалізовано розпізнавання об'єктів на шляху мобільної платформи алгоритмом YOLO, для подальшої взаємодії користувача через голосового помічника.

Розроблено інтелектуального помічника з інтеграцією на стороні API.

В ході дослідження було підтверджено ефективне використання LLM моделі, експериментально виявлено параметри які задовольняють обрані цілі, перевірено якості підтримання контексту в розмові та надання релевантних відповідей. Використання пам'яті через базу даних PostgreSQL дозволило уникнути втрати контексту, забезпечено персоналізацію помічника та підтримку психологічного комфорту, що особливо важливо для людей з обмеженими можливостями. Також проведено експериментальне дослідження здатності мобільної платформи уникати перешкод за допомогою реактивного підходу.

Отримані результати дають розуміння того, що для більш сприятливої взаємодії людини з помічником необхідно покращувати системні пропти та додавати візуалізацію запитів для уникнення недоліків моделі. У разі збільшення кількості різних типів перешкод, виникає потреба в удосконаленні алгоритму об'їзду перешкод, майбутньої інтеграції з LIDAR або розумною моделлю, для навчання робота будувати маршрут на основі вже отриманих даних.

Робота в напрямку більшої інтеграції сучасних моделей штучного інтелекту - є актуальною на даний момент та має великий потенціал для розвитку та впровадженню.

Підтримуючи проголошені резолюцією Генеральної Асамблеї Організації Об'єднаних Націй від 25 вересня 2015 року № 70/1 глобальні цілі сталого розвитку до 2030 року та результати їх адаптації з урахуванням специфіки розвитку України, що викладені у Національній доповіді «Цілі сталого розвитку: Україна», надані результати дослідження забезпечують дотримання Цілей сталого розвитку України на період до 2030 року. Вони узгоджуються з

Ціллю 3 “Забезпечення здорового способу життя та сприяння благополуччю для всіх у будь-якому віці”, зокрема з пунктом 3.9 – “ Здійснити реформу фінансування системи охорони здоров’я”, бо розроблений робот сприяє покращенню якості життя багатьом людям з обмеженими фізичними можливостями.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – К.: ДП «УкрНДНЦ», 2016. – 31 с.
2. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка освітньо-професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва»; «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2024. 57 с.
3. Стеценко К. К. Соціально-психологічні аспекти використання автономних роботів-компаньйонів в інклюзивному суспільстві // Матеріали 29-го Міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті» (16–19 квітня 2025 р., Харків). – Т. 9 : Конференція «Гуманітарні аспекти цифрового суспільства». – Харків : ХНУРЕ, 2025. – С. 54–55. – ISBN 978-966-659-413-9.
4. Бронніков А., Стеценко К. Автономний робот на Raspberry Pi з аналізом облич та емоцій в реальному часі // *Матеріали VIII Міжнародної конференції «Виробництво та мехатронні системи – M&MS 2024»* (25–26 жовтня 2024 р., Харків, Україна). – Харків : Харківський національний університет радіоелектроніки, 2024. – С. 22–23. – ISBN.
5. Yamamoto T., et al. Development of Human Support Robot as the Research Platform of a Domestic Mobile Manipulator [Електронний ресурс] // *ROBOMECH Journal*. – Режим доступу: <https://robomechjournal.springeropen.com/articles/10.1186/s400048-019-0132-3> (дата звернення: 04.07.2025).

6. Tanaka K., et al. The Pilot Study of Group Robot Intervention on Pediatric Inpatients and Their Caregivers, Using ‘New Aibo’ [Електронний ресурс] // European Journal of Pediatrics. – Режим доступу: <https://link.springer.com/article/10.1007/s0431-021-04285-8> (дата звернення: 09.07.2025).

7. Laban G., et al. Building Long-Term Human–Robot Relationships: Examining Disclosure, Perception and Well-Being Across Time [Електронний ресурс] // International Journal of Social Robotics. – Режим доступу: <https://link.springer.com/article/10.1007/s15369-026-01076-z> (дата звернення: 08.07.2025).

8. Somers J. A Revolution in How Robots Learn [Електронний ресурс] // The New Yorker. – Режим доступу: <https://www.newyorker.com/magazine/2024/12/02/a-revolution-in-how-robots-learn> (дата звернення: 10.07.2025).

9. Федасюк Д. Онтологічна модель адаптивної програмної системи для людей з когнітивними порушеннями [Електронний ресурс] // Academia.edu. – Режим доступу: <https://www.academia.edu/96348443/> (дата звернення: 09.07.2025).

10. Chen Y.-J., Jhong B.-G., Chen M.-Y. A Real-Time Path Planning Algorithm Based on the Markov Decision Process in a Dynamic Environment for Wheeled Mobile Robots [Електронний ресурс] // Applied Sciences (MDPI). – Режим доступу: <https://www.mdpi.com/2076-0825/12/4/166> (дата звернення: 08.08.2025).

11. Raspberry Pi 5 [Електронний ресурс] // Minicomp. – Режим доступу: <https://minicomp.com.ua/ua/raspberry-pi/raspberry-pi-5/raspberry-pi-5-8gb> (дата звернення: 10.08.2025).

12. Модуль Wi-Fi ESP32-CAM з камерою 2MP [Електронний ресурс] // Arduino.ua. – Режим доступу: <https://arduino.ua/prod3458-modul-wi-fi-esp32-s-kameroi-2mp> (дата звернення: 01.08.2025).

13. Ширококутова камера для Raspberry Pi 5MP 1080P з ІЧ [Електронний ресурс] // Prom.ua. – Режим доступу: <https://prom.ua/ua/p1893434133-shirokougolnaya-kamera-dlya.html> (дата звернення: 02.08.2025).

14. Модуль ESP32 WROOM 32U DevKit v4 38 Pin [Електронний ресурс] // Controller.in.ua. – Режим доступу: <https://controller.in.ua/ua/p1610820978-modul-esp32-wroom.html> (дата звернення: 02.08.2025).

15. Серводвигун SG90 2кг [Електронний ресурс] // Arduino.ua. – Режим доступу: <https://arduino.ua/prod416-servoprivod-sg90-2kg> (дата звернення: 02.08.2025).

16. Ультразвуковий датчик відстані HC-SR04 [Електронний ресурс] // Arduino.ua. – Режим доступу: <https://arduino.ua/prod182-yltrazvukovoi-datchik-rasstoyaniya-hc-sr04> (дата звернення: 02.08.2025).

17. Santana L., Maximo M., Goes L. Physical Modeling and Parameters Identification of the MG995 Servomotor [Електронний ресурс] // ResearchGate. – Режим доступу: <https://www.researchgate.net/publication/357078983> (дата звернення: 03.08.2025).

18. Tinkercad [Електронний ресурс]. – Режим доступу: <https://www.tinkercad.com/> (дата звернення: 03.08.2025).

19. PlatformIO [Електронний ресурс]. – Режим доступу: <https://platformio.org/> (дата звернення: 22.08.2025).

20. Arduino IDE [Електронний ресурс]. – Режим доступу: <https://www.arduino.cc/en/software> (дата звернення: 22.08.2025).

21. Visual Studio Code [Електронний ресурс]. – Режим доступу: <https://code.visualstudio.com/> (дата звернення: 15.08.2025).

22. Zhang X., Zeng H., Guo S., Zhang L. A Comprehensive Review of YOLO Architectures in Computer Vision (від YOLOv1 до YOLOv8) [Електронний ресурс] // MDPI. – 2023. – Режим доступу: <https://www.mdpi.com/> (дата звернення: 15.09.2025).

23. Zheng G., Liu S., Wang F., Li Z., Sun J. YOLOX: Exceeding YOLO Series in 2021 [Електронний ресурс] // arXiv. – 2021. – Режим доступу: <https://arxiv.org/> (дата звернення: 15.09.2025).

24. Bradski G. The OpenCV Library [Електронний ресурс] // Dr. Dobb's Journal of Software Tools. – 2000. – Режим доступу: <https://opencv.org/> (дата звернення: 15.09.2025).

25. Serengil S. I., Ozpinar A. LightFace: A Hybrid Deep Face Recognition Framework [Електронний ресурс] // International Journal of Computer Applications. – 2020. – Режим доступу: <https://github.com/serengil/deepface> (дата звернення: 15.09.2025).

26. Grinberg M. Flask Web Development: Developing Web Applications with Python. – O'Reilly Media, 2018. – 320 p. (дата звернення: 15.09.2025).

27. ДСТУ 3400:2015. Охорона праці. Безпека при роботі з електричними установками. – К.: Національний стандарт України, 2015. – 33 с.

28. Bouderballa A., Zegnini B., Seghier T. Determination of the Resistance of a Grounding System with Application of Finite Element Method // Przegląd Elektrotechniczny. – 2021. – №5. – С. 5–11. – DOI:10.15199/48.2021.05.02.

29. Akef S., El-Morshedy A. M. K., Samy M. M., El-Hawary H. H. Ground Resistance Estimation for a DC Converter Station // Energies. – 2024. – Vol. 17, №4, Art. 765. – DOI:10.3390/en17040765.

30. Sun P., Jiao Z., Gu H. Calculation of Short-Circuit Current in DC Distribution System Based on MMC Linearization // Frontiers in Energy Research. – 2021. – Vol. 9, Art. 634232. – DOI:10.3389/fenrg.2021.634232.

31. EasyPower. Short Circuit Currents According to IEC 60909 [Електронний ресурс]. – 07.06.2021. – 44 с. – Режим доступу: https://www.easypower.com/files/Short_Circuit_Currents_According_to_IEC_60909.pdf (дата звернення: 15.09.2025).

32. ABB. Residual Current Devices (RCDs): Whitepaper [Електронний ресурс]. – ABB Electrification, 2022. – 20 с. – Режим доступу:

https://library.e.abb.com/public/fc3d92ef8da94d38a3882b0b86afb552/ELAU_RCD_Local_Whitepaper_2022.pdf (дата звернення: 15.09.2025).

33. A Systematic Review of Large Language Model (LLM) Evaluations in Clinical Medicine [Електронний ресурс] // BMC Medical Informatics and Decision Making (Springer). – Режим доступу: <https://link.springer.com/article/10.1186/s12911-025-02954-4> (дата звернення: 15.09.2025).

34. Xu D., Lin W., Shen X., Eltarabishy S., Davis A. Intelligent Spatial Planning: A Multidimensional Study on Swiss Residential Design Using Large Language Model (GPT-4). – University College London, 2023.